GEOVANI FERREIRA DA CRUZ
GUILHERME STEINMANN

AUTORIDADE CERTIFICADORA TEMPORAL

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação da Universidade Federal de Santa Catarina

M.Sc. Juliano Romani

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Florianópolis, 2007

Trabalho Final de conclusão de curso sob o título *Autoridade Certificadora Temporal*, apresentado por Geovani Ferreira da Cruz e Guilherme Steinmann como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação, e aprovado em 29 de outubro de 2007, em Florianópolis, estado de Santa Catarina, pela banca examinadora constituída por:

Juliano Romani, M.Sc.
Orientador
Universidade Federal de Santa Catarina

Prof. Ricardo Felipe Custódio, D.Sc.
Co-orientador
Universidade Federal de Santa Catarina

Marcelo Carlomagno Carlos, M.Sc.
Universidade Federal de Santa Catarina

Túlio Cícero Salvaro de Souza, M.Sc.
Universidade Federal de Santa Catarina

Dedico este trabalho a minha família,
pela incomparável humildade, honestidade
e trabalho, por todo o esforço realizado,
e por todo apoio prestado em todas as
situações da minha vida
Geovani Ferreira da Cruz

Dedico este trabalho a minha família
que sempre me deu apoio nos momentos mais
difíceis, guiando-me para o caminho certo
e encorajando-me sempre a superar os desafios
que surgiam durante toda minha vida
Guilherme Steinmann

RESUMO

Aplicações como leilões, licitações públicas, testamentos e provas de vestibulares utilizam em seus documentos confidenciais o requisito de segurança temporalidade, que assegura a revelação do conteúdo do documento em uma data e hora pré-especificadas. Devido à evolução e popularização da informática, tem-se observado uma preferência de utilização de documentos digitais aos documentos em papel. O objetivo deste projeto é propor e implementar uma Autoridade Certificadora Temporal, capaz de prover temporalidade, além de autenticidade, integridade, não repúdio e confidencialidade aos documentos digitais tendo em vista a automatização das aplicações que empreguem temporalidade.

Palavras-chave: criptografia temporal, chaves-públicas temporal, autoridade certificadora temporal, temporalidade em documentos eletrônicos.

ABSTRACT

Some electronic documents used in applications like public biddings and wills require temporal confidentiality as a security requirement, assuring that the document's content will be only released in a specific date and time. Due to modern computer facilities, the electronic documents tend to be more used than papers documents in these kinds of applications. This work describes a public key infrastructure that provides a time key release cryptographic functionally that can envelop electronic document and control its release in the future as previously stated.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora,	p. 25
ACID	Atomicidade, Consistência, Isolamento, Durabilidade,	p. 63
ACT	Autoridade Certificadora Temporal,	p. 28
AD	Autoridade de Datação,	p. 28
API	Application Programming Interface,	p. 38
AR	Autoridade de Registro,	p. 26
ASCII	American Standard Code for Information Interchange,	p. 38
CRL	*Certificate Revogation List*,	p. 25
CSM	*Clock Secure Module*,	p. 28
CVS	*Concurrent Version System*,	p. 63
HSM	*Hardware Security Module*,	p. 26
ICP	Infra-estrutura em Chaves Públicas,	p. 27
ICPT	Infra-estrutura em Chaves Públicas Temporal,	p. 29
LabSEC	Laboratório de Segurança em Computação (LabSEC),	p. 30
LCR	Lista de Certificados Revogados,	p. 25
MVC	*Model-View-Controller*,	p. 62
PDT	*Eclipse PHP Development Tools*,	p. 62
PHP	*Hypertext Preprocessor*,	p. 30
SSL	Secure Socket Layer,	p. 30
SVN	*Subversion*,	p. 63
TLS	Transport Secure Layer,	p. 30

SUMÁRIO

1 INTRODUÇÃO

Com os grandes avanços tecnológicos das redes de computadores e com a crescente massificação do uso da internet, cada vez mais as pessoas passam a utilizar serviços informatizados, o que lhes proporciona comodidade, eficiência e redução de custos. Assim, o aumento do uso de documentos eletrônicos torna-se inerentes às necessidades cotidianas. As operações computacionais sobre documentos eletrônicos vêm sendo implementadas de modo a fornecer ao usuário a garantia de qualidade dos serviços realizados.

Com o surgimento da Certificação Digital, a utilização de documentos eletrônicos no mundo digital torna-se segura, sendo que os seguintes requisitos de segurança do documento eletrônico devem ser considerados:

- integridade: garante que o documento original não seja alterado;
- autenticidade: assegura a identidade da entidade criadora do documento;
- sigilo: impede o acesso ao conteúdo do documento por entidades não-autorizadas;
- não-repúdio: evita que a entidade transmissora negue a autenticidade das atividades efetuadas

Existe também o requisito de temporalidade em documentos eletrônicos, o qual assegura que o acesso ao conteúdo do documento somente seja possível após uma data e hora pré-especificadas, mantendo o documento em sigilo até o momento de liberação especificado. Este requisito é aplicado nos processos de enviar documentos para o futuro, onde são utilizadas as técnicas de Certificação Digital Temporal.

Este trabalho trata da implementação de um protótipo de Autoridade Certificadora Temporal, baseada em técnicas de Certificação Digital e Certificação Digital Temporal, tornando possível o envio de documentos para o futuro no mundo digital. Esta autoridade consiste numa entidade pertencente à uma infra-estrutura em chaves públicas temporal responsável por fornecer temporalidade e os demais requisitos de segurança mencionados aos documentos eletrônicos.

Primeiramente, serão apresentados conceitos acerca dos requisitos de segurança dos documentos eletrônicos e dos conceitos pertinentes a área de Segurança em Computação necessários ao entendimento de uma Autoridade Certificadora Temporal e de uma Infra-estrutura de Chaves Públicas Temporal. Na segunda parte do trabalho serão abordados os detalhes técnicos do protótipo de Autoridade Certificadora Temporal implementado.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O objetivo geral deste projeto é implementar uma Autoridade Certificadora Temporal que atenda aos requisitos de autenticidade, integridade, não-repúdio, confidencialidade e temporalidade e disponibilizá-la a acesso público para que qualquer pessoa interessada possa enviar documentos para o futuro.

1.1.2 OBJETIVOS ESPECÍFICOS

- Implementar um módulo de funções criptográficas, de certificação digital e de certificação digital temporal para PHP, necessárias à implementação da Autoridade Certificadora Temporal e também de outras autoridades;
- Implementar um protótipo Autoridade Certificadora Temporal que atenda aos requisitos de autenticidade, integridade, não-repúdio e confidencialidade e temporalidade, provendo melhorias em relação as vulnerabilidades dos sistemas implementados;
- Disponibilizar a acesso público o projeto desenvolvido, de modo que qualquer pessoa interessada em enviar documentos para o futuro possa realizá-lo através do sistema a ser implementado, com toda a segurança necessária.

1.2 JUSTIFICATIVA

A mesma evolução tecnológica que traz junto consigo a eficiência, comodidade e redução de custos por parte de seus usuários, acaba criando grandes problemas. Juntamente com esse crescimento desenfreado surge um número muito grande de usuários mal-intencionadas que procuram explorar as vulberabilidades existentes das estruturas das redes de computadores atuais para obter benefício próprio. Essas pessoas podem ser estudantes, com fins de aprendizado, que testam o nível de segurança dessas estruturas, empresas querendo descobrir as estratégias da concorrência, terroristas tentando captar informações importantes para seus ataques, ladrões tentando fazer desvios de dinheiro para suas contas através de transações bancárias, consumidores tentando negar uma dívida, dentre diversos outros. A falta de segurança presente nas estruturas atuais acrescido ao fato de que essa evolução tecnológica tende a aumentar cada vez mais acaba por gerar uma necessidade de desenvolvimento de diversas soluções em segurança para atender às necessidades desse mercado dinâmico e em expansão.

Ao mesmo tempo que se cria uma expectativa ou até mesmo uma necessidade de substituição de documentos em papel por documentos eletrônicos, vários estudos tem sido realizados na área de segurança em computação para solucionar os problemas comentados. Existe uma preferência muito forte sob os processos automatizados, que de alguma forma, acabem reduzindo a burocracia existente sobre os documentos tradicionais em formato papel. Essa burocracia, por sua vez, faz parte da cotidiano de um número extremamente grande de pessoas, tanto físicas quanto jurídicas, obrigando-as a trabalharem com os mais variados tipos de documentos, necessários à realização de uma série de atividades de fins jurídicos, contábeis, comerciais, financeiros, econômicos, ou até mesmo pessoais. Não é difícil de imaginar que seria preferível realizar todas estas atividades atualmente estruturadas sob documentos em papel através do uso do computador sob sob o formato de documento eletrônicos. Embora muito se tenha feito em prol da automatização destas tarefas, infelizmente, ainda persistem um número muito grande de atividades que não são possíveis de serem realizadas digitalmente.

Um nicho de aplicações que ainda não têm um suporte eletrônico completo são aquelas que necessitem enviar informações para o futuro. Podemos classificar neste nicho processos como: licitações públicas, provas de vestibular, autuações, testamentos, informações governamentais, depósito legal de documentos, pagamento eletrônico, dentre outros. Em todos se é necessário um documento em papel geralmente autenticado, assinado a próprio punho e lacrado, que somente poderá ser aberto após uma determinada data e hora. Isso acaba gerando custos de cartório, impressão, transporte e, até mesmo de disponibilização de um local para a realização formal da cerimônia de abertura dos documentos lacrados. Através de um le-

vantamento bibliográfico realizado, verificou-se que ainda não existe em funcionamento, uma solução computacional segura capaz de provêr o envio de documentos para o futuro.

A motivação deste trabalho consiste na importância do desenvolvimento de uma solução para um problema atual, de grande abrangência, e de muito interesse por parte de pesquisadores na área de segurança em computação, das entidades que irão fornecer os serviços de enviar documentos digitais para o futuro e, principalmente, dos usuários destes serviços.

1.3 DEFINIÇÃO DO PROBLEMA

Para facilitar o entendimento, primeiramente será feito um detalhamento de um dos processos que necessita enviar documentos para o futuro, a licitação pública, para o levantamento dos problemas a serem solucionados, que consiste nos requisitos de segurança a serem considerados na implementação do protótipo de Autoridade Certificadora Temporal.

O processo de licitação pública divide em modalidades como leilão, pregão, tomada de preços, concorrência, dentre outras. Uma maior abordagem acerca deste processo pode ser consultada em (PEREIRA; CUSTÓDIO; NOTOYA, 2003). Abaixo, as fases de realização de uma licitação pública:

1. Preparação: A entidade pública desenvolve o Edital referente à modalidade de compra desejada e o divulga aos interessados através do meio de comunicação apropriado.

2. Habilitação: Os interessados entregam a proposta à entidade pública em um envelope lacrado. Essa proposta contém as especificações do objeto de venda, bem como os demais documentos necessários.

3. Julgamento: É realizada uma sessão pública, em data e hora especificadas no Edital do processo, onde são abertas as propostas dos fornecedores. A entidade pública seleciona a melhor proposta, através dos critérios de avaliação divulgados no Edital do processo.

Fazendo uma avaliação deste processo, onde os documentos estão em formato não-digital, os requisitos de segurança mencionados são assegurados. O requisito de temporalidade (acesso ao conteúdo somente na data e hora previstas) é garantido através da realização da sessão pública. A integridade do documento é verificada ao não se encontrar rasuras (modificações, colagens, partes do documento riscadas, rasgos ou sobrescritos) no documento de papel. O não-repúdio e a autenticidade são garantidos através da assinatura esferográfica ou do carimbo do fornecedor. O sigilo, por sua vez, é alcançado através da verificação do lacre do envelope

contendo os documentos (uma vez que o lacre esteja intacto, é possível garantir que ninguém teve acesso ao conteúdo do documento).

Para que, por exemplo, o processo de licitação pública possa ser realizado digitalmente, pelo menos a mesma credibilidade sobre o processo deve ser mantida em se tratando de segurança. Realizar este processo em meio digital traz várias vantagens às entidades envolvidas neste processo. Para a entidade pública, os custos com a infra-estrutura necessária para a realização da sessão pública (local, funcionários, organização) seriam praticamente eliminados e o processo se torna mais eficiente e prático. Para os fornecedores, eliminar-se-iam os custos com papel e impressão e haveria uma maior garantia contra fraudes do sistema público, uma vez que as propostas seriam avaliadas automaticamente pelo sistema informatizado do processo.

Com o surgimento das técnicas de Certificação Digital e Certificação Digital Temporal, a segurança dos documentos digitais são asseguradas. A utilização de funções hash garante a integridade do documento digital e as assinaturas digitais utilizadas em conjunto com funções hash garantem a autenticidade (STINSON, 1995). O não-repúdio dos documentos é alcançado com o uso da criptografia assimétrica, através da utilização de pares de chaves correspondentes (uma pública e uma privada) para cifrar e decifrar os documentos (AUSTRALIA; CAELLI; LITTLE, 1995). O sigilo dos documentos é alcançado através dos algoritmos de cifragem e decifragem (SCHNEIER, 1995). E, para finalizar, (RIVEST; SHAMIR; WAGNER, 1996) sugere o uso de uma terceira entidade confiável para prover a temporalidade.

Em prol de tornar viável a implementação de operações informatizadas sobre documentos digitais que necessitem enviar documentos para o futuro, com a mesma segurança, ou até maior que nos documentos em papel os requisitos de segurança devem ser considerados. Uma vez que não existem soluções completas nesse sentido, o problema é implementar uma Autoridade Certificadora Temporal que se utilize essas técnicas para prover os requisitos de segurança.

1.4 ESTRUTURA DO TRABALHO

No capítulo 2 são abordados os conceitos necessários de criptografia bem como os conceitos de criptografia temporal. A Infra-estrutura em Chaves Públicas é abordada e os módulo de hardware (HSM e CSM) também são abordados.

O módulo de funções criptográficas para PHP *Modcryptosec*, que foi implementado com o objetivo de disponibilizar funções criptográficas para a linguagem PHP é apresentado no capítulo 3. As funções implementadas no módulo são apresentadas neste capítulo.

No capítulo 4 o projeto de uma Autoridade Certificadora Temporal é apresentado, bem como os módulos necessários para o funcionamento desta autoridade.

O capítulo 5 apresenta o protótipo da Autoridade Certificadora Temporal implementado e as tecnologias utilizadas para a implementação.

Por fim, são apresentadas as considerações finais, as contribuições do trabalho e são descritos os trabalhos futuros.

2 REVISÃO

De acordo com (CUSTÓDIO et al., 2006), para manter o sigilo dos documentos em um processo temporal, três aspectos principais devem ser considerados. Primeiramente, deve-se garantir a confidencialidade do documento durante o processo de envio. O segundo deve prever que o conteúdo do documento seja armazenado de forma confidencial. O terceiro aspecto deverá garantir que o conteúdo do documento possa ser visualizado pelas entidades autorizadas somente após uma data e hora especificados.

Para garantir a confidencialidade do documento durante o processo de envio, deve-se assegurar que ninguém durante o processo que por ventura intercepte o documento, possa ter o conhecimento de seu conteúdo. No caso de documentos em papel, um envelope lacrado transportará esses documentos até o destino. O lacre garante que o conteúdo do documento não foi violado. No caso de documentos eletrônicos, técnicas de criptografia são utilizadas para cifrar o conteúdo do documento, ou seja, o conteúdo do documento original é criptografado fazendo que pareça sem sentido para um interceptador.

> "A criptografia pode ser entendida como um conjunto de métodos e técnicas para cifrar ou codificar informações legíveis por meio de um algoritmo, convertendo um texto original em um texto ilegível, sendo possível mediante o processo inverso recuperar as informações originais" (MORENO; PEREIRA; CHIARAMONTE, 2005 apud SIMON, 1999).

No caso do segundo aspecto, para garantir o armazenamento do documento de forma confidencial, deve-se prever que ninguém, além dos destinatários, possa visualizar o conteúdo do documento. Em documentos em papel o lacre deve ser aberto somente pelos destinatários, isto é, somente os destinatários têm a permissão para abrir o lacre. Em documentos eletrônicos as técnicas de criptografia fazem com que somente as pessoas que possam decifrar o documento tenham acesso ao conteúdo.

O último aspecto, que trata da garantia que o documento possa ser visualizado pelos destinatários somente após um horário especificado, deve assegurar que o documento somente seja aberto após a data e a hora especificados pelo(s) autor(es). Para documentos em papel uma aber-

tura pública do lacre no horário previsto assegura que o conteúdo não foi anteriormente visualizado. Para documentos eletrônicos as técnicas de criptografia temporal podem ser utilizadas para que o conteúdo do documento somente possa ser decifrado no tempo pré-determinado., isto é, somente os destinatários têm a permissão para abrir o lacre

2.1 CONCEITOS DE CRIPTOGRAFIA

Segundo (MORENO; PEREIRA; CHIARAMONTE, 2005) o ato de transformar um texto legível (original) em um texto ilegível dá-se o nome de cifrar (criptografar, codificar, encriptar). O processo inverso dá-se o nome de decifrar (decriptografar, decodificar, decriptar).

O procedimento de transformação do texto em legível ou ilegível é realizado por um algoritmo de criptografia que consiste numa seqüência de procedimentos matemáticos capaz de cifrar ou decifrar. Além do algoritmo, uma chave de criptografia protege a informação cifrada. A chave é um número ou um conjunto de números, que é única, e alimenta o algoritmo de criptografia (MORENO; PEREIRA; CHIARAMONTE, 2005).

Os algoritmos de criptografia podem ser classificados em algoritmos de chave simétrica e algoritmos de chave assimétrica.

2.1.1 ALGORITMOS DE CHAVE SIMÉTRICA

Os algoritmos de chave simétrica utilizam a mesma chave para cifrar e para decifrar uma informação, ou seja, em um processo de envio de uma mensagem cifrada tanto o remetente (que cifra a mensagem para o envio) quanto o destinatário (que decifra a mensagem recebida) utilizam a mesma chave no algoritmo de criptografia.

A vantagem desse algoritmo é a rapidez nas operações de criptografia, porém existe o "problema da distribuição das chaves" (MORENO; PEREIRA; CHIARAMONTE, 2005), em que a chave deve ser distribuída entre todos os destinatários para que possam decifrar a mensagem, o que causa perda de tempo e possibilita que usuários não autorizados tenham acesso à chave. Caso alguém não autorizado tiver acesso à chave, o conteúdo das mensagens poderá ser revelado prejudicando o sigilo da comunicação.

2.1.2 ALGORITMOS DE CHAVE ASSIMÉTRICA

Devido ao problema de distribuição de chaves, em 1976 Whitfield Diffie e Martin Hellman inventaram a criptografia de chaves públicas (MORENO; PEREIRA; CHIARAMONTE, 2005),

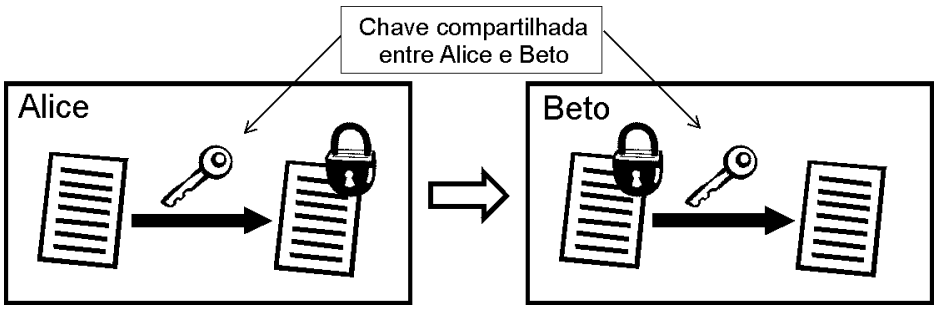

Figura 1: Exemplo de envio de um documento de Alice para Beto utilizando a criptografia simétrica.

também chamada de criptografia de chave assimétrica. Nesse tipo de criptografia cada pessoa possui um par de chaves, uma pública e uma privada diferentes. O par de chaves é um conjunto com dois números primos gigantescos fatorados entre si (SILVA, 2004). A chave pública é divulgada enquanto a chave privada é mantida em segredo.

Num processo de envio de uma mensagem cifrada por criptografia assimétrica, o remetente cifra sua mensagem com a chave pública do destinatário. Ao receber a mensagem, o destinatário, de posse da sua chave privada, decifra a mensagem.

Com esse algoritmo o problema de distribuição de chaves é implementado (MORENO; PEREIRA; CHIARAMONTE, 2005). No entanto esse tipo de criptografia é mais lento quando se comparado à criptografia simétrica.

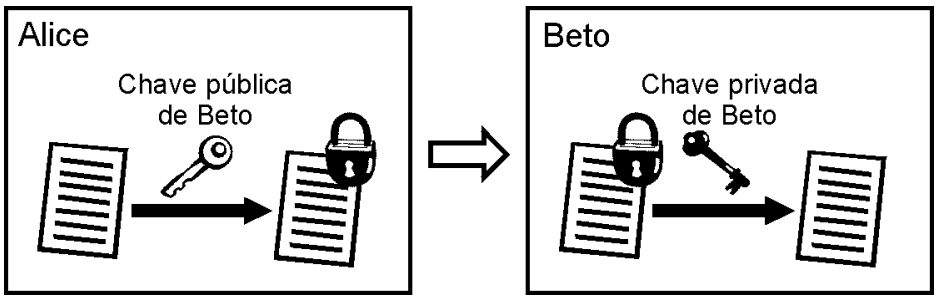

Figura 2: Exemplo de envio de um documento de Alice para Beto utilizando a criptografia assimétrica.

Segundo (SILVA, 2004), a segurança da criptografia assimétrica está baseada no fator tempo, ou seja, atualmente seria necessário muito tempo (podendo ultrapassar uma centenas de anos) para derivar o número e conhecimento do par dos números primos do par de chaves.

2.1.3 ASSINATURA DIGITAL

A assinatura digital é um dos recursos de segurança providos pela criptografia assimétrica (SILVA, 2004). Ela consiste em provar que um remetente realmente enviou uma mensagem,

assim como uma assinatura em um documento de papel.

"A assinatura digital é uma mensagem que só uma pessoa poderia produzir, mas que todos possam verificar" (MORENO; PEREIRA; CHIARAMONTE, 2005). Isso assegura que o remetente da mensagem será responsável pelo seu conteúdo, não podendo alegar que a mensagem foi forjada, garantindo assim o não-repúdio da mensagem.

> "O fundamento da assinatura digital está baseado no par de chaves pública e privada. Dessa forma deve existir uma chave privada que somente a entidade que assinou conhece e uma chave pública que irá servir de base para a verificação dessa assinatura" (SILVA, 2004).

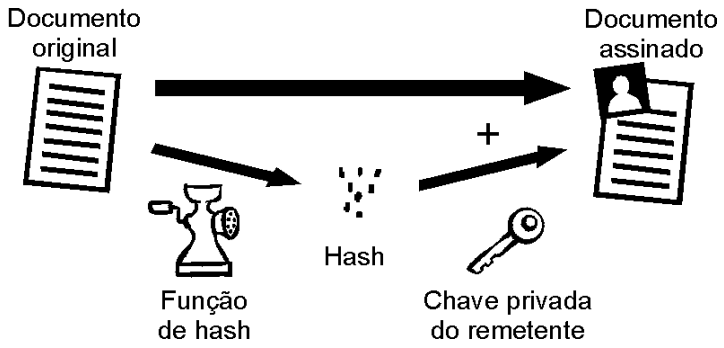

Figura 3: Exemplo de assinatura de um documento.

Na prática, o processo de assinatura digital consiste em:

1. O remetente calcula o *hash* de uma mensagem e cifra este *hash* calculado com sua chave privada;
2. O remetente envia a mensagem e o *hash* cifrado ao destinatário;
3. O destinatário recebe a mensagem e o *hash* cifrado do remetente;
4. O destinatário, de posse da chave pública decifra o *hash* recebido;
5. O destinatário calcula o *hash* da mensagem recebida;
6. O destinatário compara os dois *hashes*. Se forem iguais significa que somente a chave privada do remetente pode ter produzido o hash cifrado da mensagem, garantindo o não-repúdio da mensagem por parte do remetente.

Segundo (SILVA, 2004), a assinatura digital também assegura a integridade da assinatura digital pois se houve alguma modificação durante o envio da mensagem, a assinatura digital passa a ser inválida uma vez que tinha validade para o texto original.

2.1.4 CERTIFICADO DIGITAL

Certificados digitais garantem que uma chave pública pertença realmente a uma entidade (pessoa, empresa, computador, etc.). Com o uso dos certificados digitais, o remetente tem como provar que a chave pública do destinatário pertença realmente ao destinatário, garantindo a autenticidade. No caminho inverso, o destinatário pode provar que a chave pública disponibilizada realmente pertença ao remetente da mensagem.

Para garantir a autenticidade dos certificados, uma terceira entidade confiável deve assegurar sua veracidade, assinando digitalmente estes certificados. À essa terceira entidade confiável dá-se o nome de Autoridade Certificadora (AC).

Cada certificado digital possui uma data de validade que pode expirar (SILVA, 2004). Ao expirar um certificado (seja por período de validade ou por troca de atributos em sua estrutura) a AC envia este certificado para uma *Certificate Revogation List* (CRL) ou Lista de Certificados Revogados (LCR). Os certificados contidos numa CRL são considerados inválidos (SILVA, 2004).

2.1.5 AUTORIDADE CERTIFICADORA (AC)

Uma AC é uma entidade responsável por emitir certificados digitais (SILVA, 2004), garantindo assim a veracidade destes certificados. Esses certificados emitidos possuem a assinatura digital da AC que o emitiu que, graças à idoneidade desta AC, é reconhecida normalmente como confiável (SILVA, 2004).

Os certificados emitidos por uma AC seguem o formato X.509 [RFC4523] sendo que existe uma hierarquia das ACs em forma de árvore onde existe uma AC, pertencente ao maior nível da hierarquia, responsável por autorizar outras ACs que assinam os certificados de entidades (podendo esta ser uma pessoa, empresa ou máquina) ou de outras autoridades.

Como o certificado possui a assinatura de uma AC, é possível imaginar que uma relação de confiança pode ser formada uma vez que uma AC pode assinar o certificado de uma outra AC e assim conseqüentemente até a assinatura do certificado da entidade (SILVA, 2004).

Nesta relação de confiança pode-se distinguir três tipos de ACs que pertencem à uma hierarquia em forma de árvore: as AC-raiz, AC-intermediárias e as AC-finais.

As AC-raízes são ACs que estão no topo da relação de confiança e que são responsáveis por emitir certificados para outras ACs. Os certificados deste tipo de AC são auto-assinados, ou seja, elas próprias assinam seu certificado pois não são subordinadas a nenhuma outra AC.

As AC-intermediárias são ACs também são responsáveis por emitir certificados digitais para outras ACs, porém este tipo de autoridade é subordinado à uma AC-raiz, ou seja, seu certificado foi assinado por uma AC-raiz ou uma outra AC-intermediária superior na relação de confiança à ela.

Já as AC-finais são autoridades certificadoras que emitem certificados somente para usuários finais. Como nas AC-intermediárias, a AC-final é subordinada a uma AC-raiz ou uma AC-intermediária.

Um problema que não é difícil de imaginar, é quem deve ser responsável por gerenciar esta AC. Em alguns países o próprio governo a controla, porém é fato que nem todas as pessoas confiam no governo de seu país. Para resolver este problema é necessário criar uma infra-estrutura destas autoridades, permitindo ao usuário dessa infra-estrutura escolher a AC de sua maior confiança.

2.1.6 AUTORIDADES DE REGISTRO (AR)

Uma Autoridade de Registro (AR) "pode ter duas atribuições bem definidas: verificar o conteúdo do certificado para uma AC e fornecer os mecanismos para ingresso de novos usuários na AC" (SILVA, 2004).

Uma Autoridade de Registro (AR) é uma entidade responsável por validar os dados dos usuários, sejam estes pessoas físicas, jurídicas, softwares, ou mesmo outras ACs e ARs. No caso de inexistência de uma AR, as ACs podem executar as tarefas da mesma embora que, geralmente, essas autoridades trabalhem em conjunto.

Após avaliar os dados de um usuário, a AR submete uma requisição a AC que está credenciada, solicitando a geração do certificado digital para o usuário. Para que a requisição possa ser reconhecida pela AC, ela deve ser assinada por uma AR credendiada por esta AC, isto é, a AR precisa estar na lista de ARs confiávies da AC (SILVA, 2004).

2.1.7 MÓDULO DE HARDWARE SEGURO

(DIAS et al., 2004) definem um módulo de hardware seguro, ou em inglês *Hardware Security Module* (HSM), como um dispositivo de hardware, com poder de processamento e armazenagem, específico para executar serviços tais como a geração e o armazenamento de chaves, realização de operações criptográficas, em geral, sobre aplicações que exigem um elevado grau de segurança. O HSM gera, usa e destrói as chaves criptográficas não permitindo que entidades externas tenham acesso as mesmas.

2.1.8 INFRA-ESTRUTURA EM CHAVES PÚBLICAS (ICP)

As ACs e ARs estão organizadas numa infra-estrutura denominada Infra-estrutura em Chaves Públicas (ICP) que consiste num conjunto de serviços e procedimentos para gerenciar e prover o uso de certificados digitais. Para (SILVA, 2004), uma ICP é um ambiente que possibilita que as transações feitas no mundo digitais tenham os mesmos resultados do que fora do mundo digital.

Segundo (SILVA, 2004), "a infra-estrutura em chaves públicas oferece vários serviços, que podem ou não ser utilizados, dependendo da política de segurança adotada na empresa e do comprometimento dos usuários dos certificados digitais". Dentre os principais serviços oferecidos estão o sigilo, a autenticidade, a integridade e o não-repúdio.

2.2 CONCEITOS DE CRIPTOGRAFIA TEMPORAL

A criptografia temporal é responsável por tratar o envio de informações para o futuro. Para (RIVEST; SHAMIR; WAGNER, 1996), o objetivo da criptografia temporal é cifrar um documento eletrônico que não possa ser decifrado por ninguém, nem mesmo pelo autor do documento, até que um determinado período de tempo tenha passado.

Na realidade, a criptografia temporal trata dos mesmos requisitos da criptografia de dados adicionando o requisito de temporalidade. As informações enviadas para o futuro devem ser transportadas e armazenadas de forma segura com a garantia de liberação na data e hora previstas.

(RIVEST; SHAMIR; WAGNER, 1996) propõem duas técnicas de implementar a temporalidade de documentos digitais: através de um quebra-cabeça computacional (*Time-lock puzzle*) e através do uso de uma terceira entidade confiável.

A temporalidade por quebra-cabeça computacional consiste na realização de uma seqüência de operações matemáticas, não escalonáveis e não distribuídas, aonde o tempo total de processamento dessas operações é aproximadamente o tempo que a informação deve permanecer em sigilo. Ao finalizar a seqüência de operações, a informação é descoberta. Torna-se inviável para o escopo deste projeto uma vez que o método não garante uma precisão no tempo de liberação da informação e porque é necessário um hardware dedicado para processar as operações, ainda desconsiderando a possibilidade de falhas do hardware.

Uma segunda maneira de se obter a temporalidade é através de uma terceira entidade confiável, a qual se compromete em manter o sigilo da informação até o seu tempo de liberação.

A esta terceira entidade dá-se o nome de Autoridade Certificadora Temporal.

2.2.1 AUTORIDADE CERTIFICADORA TEMPORAL (ACT)

Uma Autoridade Certificadora Temporal (ACT) é responsável por emitir certificados digitais temporal, o qual contém uma chave de sigilo que pode ser utilizada pelos usuários para cifrarem seus documentos. Esses documentos permanecem cifrados até a data da liberação de uma chave de liberação correspondente pela ACT. De posse da chave de liberação correspondente é possível decifrar o documento.

Assim como nas ACs, os certificados emitidos por uma ACT seguem o padrão X.509. O certificado de uma ACT deve ser assinado por uma AC estabelecendo uma relação de confiança. Fazendo uma rápida análise, uma ACT possui as mesmas características de uma AC-final, adicionada de requisitos de temporalidade.

Para que a ACT possa ter seu relógio fornecendo uma data e hora confiáveis, este deve ser sincronizado com o relógio de uma Autoridade de Datação. Portanto, a segurança da informação é garantida pela ACT pelo uso das técnicas de criptografia assimétrica (certificados digitais, assinaturas digitais, chaves de sigilo e de liberação correspondentes) e a temporalidade pela confiança atribuída a Autoridade de Datação.

2.2.2 AUTORIDADES DE DATAÇÃO (AD)

Uma Autoridade de Datação (AD) consiste numa entidade confiável responsável por fornecer a data e hora corretas para as demais entidades interessadas, que confiam na imparcialidade desta Autoridade (DIAS; CUSTÓDIO; DEMÉTRIO, 2003). É composta basicamente, de um relógio que deve estar sincronizado com uma fonte confiável de tempo.

2.2.3 MÓDULO DE RELÓGIO SEGURO (CSM)

Algumas aplicações de criptografia temporal exigem um elevado grau de segurança e/ou de desempenho. Tais aplicações podem utilizar um hardware específico, desenvolvido para atender estes requisitos, como um Módulo de Relógio Seguro, ou *Clock Secure Module* (CSM), em inglês. Este módulo é um dispositivo de hardware projetado para executar funções de criptografia temporal tais como carimbo de tempo, geração do par de chaves criptográficas e armazenamento de chaves, sejam estas de ACTs ou de servidores de carimbo de tempo.

Ainda não existe no mercado um CSM com as características apresentadas. Trata-se de

um projeto em desenvolvimento pelo LabSEC (Laboratório de Segurança em Computação) da Universidade Federal de Santa Catarina. Esse projeto é um dos objetivos da dissertação de mestrado do mestrando Juliano Romani que visa implementar esse dispositivo de modo a realizar a sincronização segura de relógio. A Figura 4 mostra um exemplo do serviço de fornecimento de hora provido por uma AD através de um servidor de carimbo de tempo, bem como uma possível arquitetura de utilização do CSM.

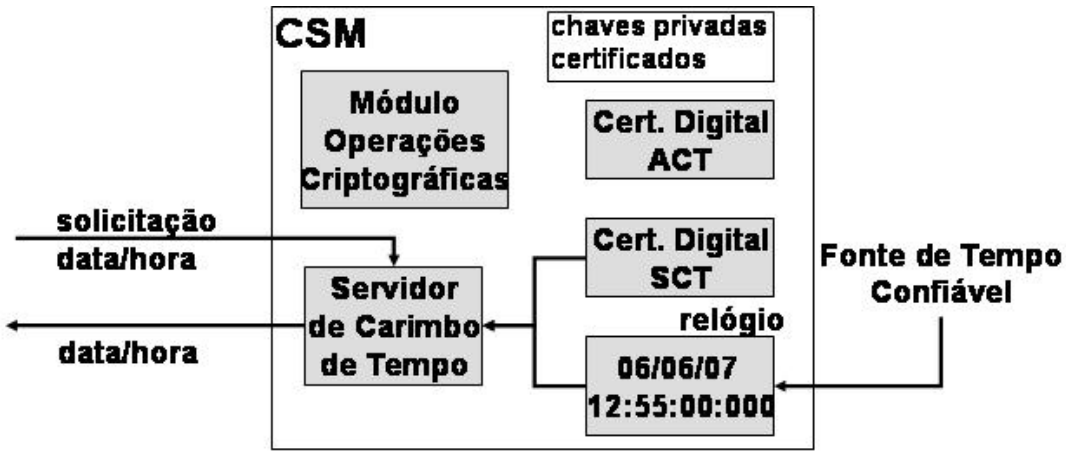

Figura 4: Exemplo de CSM e o serviço de fornecimento de tempo.

2.2.4 INFRA-ESTRUTURA EM CHAVES PÚBLICAS TEMPORAL (ICPT)

Uma Infra-estrutura em Chaves Públicas Temporal (ICPT) fornece os mesmos serviços disponíveis numa Infra-estrutura de Chaves Públicas, adicionada dos serviços de criptografia temporal e dos elementos que a compõem tais como as ADs e as ACTs. Essa infra-estrutura consiste nessa arquitetura de componentes necessários para prover os serviços de criptografia temporal aos usuários.

3 *MODCRYTPOSEC*

Após realizada a etapa de revisão bibliográfica, foi necessário escolher a linguagem de programação para implementar o protótipo da Autoridade Certificadora Temporal. A linguagem de programação *Hypertext Preprocessor* (PHP) foi escolhida, devido a facilidade de uso e o suporte existente para essa linguagem no Laboratório de Segurança em Computação (LabSEC) (LabSEC), laboratório de pesquisa que propôs o tema deste trabalho. Diversas pessoas do LabSEC já dominavam esta linguagem de programação e já haviam desenvolvido outros projetos na área de criptografia de dados nesta linguagem.

Escolhida a linguagem de programação, verificou-se que o PHP ainda não possuía nenhuma biblioteca específica para este tipo de aplicação. Outros projetos em desenvolvimento pelo laboratório também estavam sendo desenvolvidos em PHP e também precisavam de uma biblioteca de funções criptográficas. Com isso surgiu a necessidade de criação de uma biblioteca de funções de criptográficas para PHP, necessárias não somente a implementação do protótipo da Autoridade Certificadora Temporal, mas também aos outros projetos do LabSEC. Um projeto do módulo foi desenvolvido visando atender essas necessidades, o qual foi implementado pelos autores do projeto. Este módulo foi chamado Modcryptosec.

A Modcryptosec é um módulo de funções criptográficas para PHP, implementada na linguagem de programação C++ sobre uma biblioteca de funções existente em C++, a Libcryptosec. O módulo visa suprir as necessidades dos usuários em PHP através da utilização da Libcryptosec, realizando as conversões necessárias. Nas próximas seções, serão abordados os detalhes técnicos da implementação da Modcryptosec.

3.1 ARQUITETURA DA MODCRYPTOSEC

Recentemente, o LabSEC desenvolveu uma biblioteca de funções chamada Libcryptosec, que é uma extensão do OpenSSL (a biblioteca de funções criptográficas mais utilizada na área de Segurança em Computação). A biblioteca OpenSSL implementa os protocolos Secure Socket Layer (SSL) e Transport Secure Layer (TLS) na linguagem C, disponibilizando várias funções de criptografia (OPENSSL, 2007).

A Libcryptosec foi implementada em C++ com orientação a objetos, objetivando facilitar e flexibilizar o desenvolvimento de soluções e aplicações em criptografia de dados.

A Modcryptosec foi desenvolvida em C++, utilizando-se da Libcryptosec. A Figura 5 mostra a arquitetura do módulo implementado.

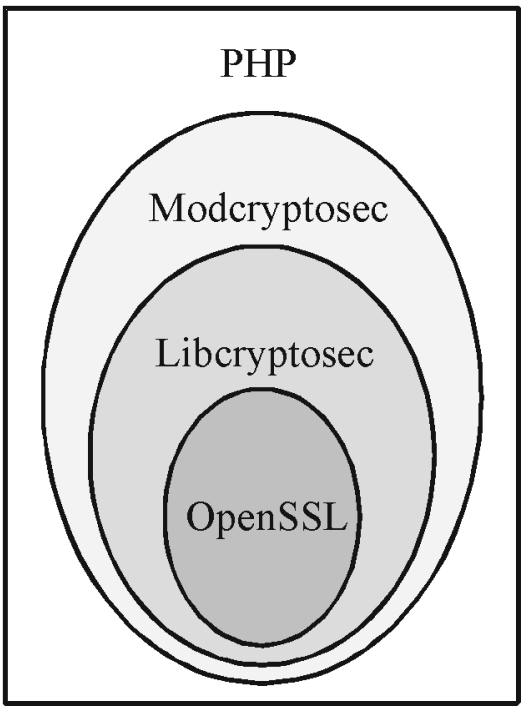

Figura 5: Arquitetura da Modcrytosec.

3.2 DESCRIÇÃO DAS FUNÇÕES DA MODCRYPTOSEC

O módulo contém sessenta e nove funções, classificadas nas seguintes categorias:

- Certificados
- Extensões de Certificados
- Lista de Certificados Revogados (LCR)
- Extensões de LCR
- Requisições de Certificados
- Extensões de Requisições
- Emissão de Certificados, Requisições e LCR

- Engines (HSM ou CSM)
- Criptografia

Abaixo, segue uma descrição de cada categoria de funções e as assinaturas das funções implementadas.

3.2.1 CERTIFICADOS

As funções da categoria certificados permitem a manipulação de certificados no formato X.509 existentes, possibilitando a extração dos principais campos deste certificado, ou mesmo verificar a assinatura digital do certificado. Abaixo, as funções implementadas:

1. *libcryptosec_certificate_get_subject($certData) : array*
2. *libcryptosec_certificate_get_issuer($certData) : array*
3. *libcryptosec_certificate_get_serial_number($certData) : int*
4. *libcryptosec_certificate_get_not_after($certData) : int*
5. *libcryptosec_certificate_get_not_before($certData) : int*
6. *libcryptosec_certificate_get_version($certData) : int*
7. *libcryptosec_certificate_get_message_digest_algorithm($certData) : string*
8. *libcryptosec_certificate_get_public_key($certData) : string*
9. *libcryptosec_certificate_verify_signature_with_issuer($certData, $certIssuerData) : bool*
10. *libcryptosec_certificate_verify_signature_with_public_key($certData, $issuerPublicKey) : bool*
11. *libcryptosec_certificate_get_fingerprint($certData, $algorithm="SHA1") : string*

3.2.2 EXTENSÕES DE CERTIFICADOS

Permitem a extração dos campos das extensões dos certificados X.509 de um determinado certificado.

1. *libcryptosec_certificate_extension_get_key_usage($certData) : array*

2. *libcryptosec_certificate_extension_get_basic_constraints($certData) : array*
3. *libcryptosec_certificate_extension_get_subject_key_identifier($certData) : array*
4. *libcryptosec_certificate_extension_get_authority_key_identifier($certData) : array*
5. *libcryptosec_certificate_extension_get_extended_key_usage($certData) : array*
6. *libcryptosec_certificate_extension_get_issuer_alternative_name($certData) : array*
7. *libcryptosec_certificate_extension_get_subject_alternative_name($certData) : array*
8. *libcryptosec_certificate_extension_get_certificate_policies($certData) : array*
9. *libcryptosec_certificate_extension_get_crl_distribution_points($certData) : array*
10. *libcryptosec_certificate_extension_get_unknown_extensions($certData) : array*
11. *libcryptosec_certificate_extension_get_extensions($certData) : array*

3.2.3 LISTA DE CERTIFICADOS REVOGADOS

Permitem a manipulação das listas de certificados revogados, possibilitando a revogação dos certificados.

1. *libcryptosec_crl_get_issuer($crlData) : array*
2. *libcryptosec_crl_get_this_update($crlData) : int*
3. *libcryptosec_crl_get_next_update($crlData) : int*
4. *libcryptosec_crl_get_revoked_certificates($crlData) : array*
5. *libcryptosec_crl_verify_signature($crlData, $certIssuer) : boolean*

3.2.4 EXTENSÕES DE LISTA DE CERTIFICADOS REVOGADOS

Permitem a manipulação das extensões das listas de certificados revogados, complementando a categoria 3.2.3.

1. *libcryptosec_crl_extension_get_crl_number($crlData) : array*
2. *libcryptosec_crl_extension_authority_key_identifier_get_key_Identifier($crlData) : int*

3. *libcryptosec_crl_extension_get_unknown_extensions($crlData) : int*

4. *libcryptosec_crl_extension_get_extensions($crlData) : array*

3.2.5 REQUISIÇÕES DE CERTIFICADOS

Permitem a manipulação de requisições de certificados X.509.

1. *libcryptosec_request_get_subject($reqData) : array*

2. *libcryptosec_request_get_version($reqData) : int*

3. *libcryptosec_request_get_public_key($reqData) : string*

4. *libcryptosec_request_verify_signature($reqData) : boolean*

3.2.6 EXTENSÕES DE REQUISIÇÕES

Permitem a manipulação das extensões das requisições de certificados X.509, complementando a categoria 3.2.5.

1. *libcryptosec_request_extension_get_unknown_extensions($reqData) : array*

2. *libcryptosec_request_extension_get_extensions($reqData) : array*

3.2.7 EMISSÃO DE CERTIFICADOS, REQUISIÇÕES E LISTA DE CERTIFICADOS REVOGADOS

Estas funções permitem a criação de certificados X.509, criação de requisições e criação de listas de certficados revogados.

1. *libcryptosec_generate_request($subject, $publicKey, $privateKey, $password=null, $extensions = array()) : string*

2. *libcryptosec_generate_request_with_engine($subject, $publicKey, $enginePath, $engineID, $keyName, $commands = null, $extensions = array()) : string*

3. *libcryptosec_issue_certificate($serialNumber, $issuer, $subject, $notBefore, $notAfter, $certPublicKey, $authorityPrivateKey, $password=null, $extensions = array()) : string*

4. *libcryptosec_issue_certificate_with_engine($serialNumber, $issuer, $subject, $notBefore, $notAfter, $certPublicKey, $enginePath, $engineID, $keyName, $commands = null, $extensions = array()) : string*

5. *ilibcryptosec_issue_crl($serialNumber, $issuer, $lastUpdate, $nextDate, $revokedCertificates, $authorityPrivateKey, $password=null, $extensions = array()) : string*

6. *libcryptosec_issue_crl_with_engine($serialNumber, $issuer, $lastUpdate, $nextDate, $revokedCertificates, $enginePath, $engineID, $keyName, $commands = null, $extensions = array()) : string*

3.2.8 ENGINES (HSM OU CSM)

Permite a utilização dos hardwares criptográficos.

1. *libcryptosec_engine_test_init($enginePath, $engineID) : bool*

2. *libcryptosec_engine_get_available_commands($enginePath, $engineID) : array*

3.2.9 CRIPTOGRAFIA

Nesta categoria, as funções disponibilizam as principais funções de criptografia, de cifragem e decifragem simétricas e assimétricas, de assinatura digital, de funções hash, dentre outras.

1. *libcryptosec_crypto_sign($hash, $signAlgorithm, $keyAlgorithm, $privateKey, $passwd = NULL) : stream*

2. *libcryptosec_crypto_sign_with_engine($hash, $signAlgorithm, $enginePath, $engineID, $keyName, $commands = NULL) : stream*

3. *libcryptosec_crypto_verify_signature($signature, $hash, $signAlgorithm, $keyAlgorithm, $publicKey) : boolean throws Exception*

4. *libcryptosec_crypto_get_public_key_from_engine($enginePath, $engineID, $keyName, $commands = NULL) : array*

5. *libcryptosec_crypto_asymmetric_key_get_size_from_engine($enginePath, $engineID, $keyName, $commands = NULL) : int*

6. *libcryptosec_crypto_asymmetric_key_get_algorithm_from_engine($enginePath, $engineID, $keyName, $commands = NULL) : string*

7. *libcryptosec_crypto_get_public_key($privateKey, $passwd = null) : array throws Exception*
8. *libcryptosec_crypto_asymmetric_key_string_get_size($privateKey, $passwd = null) : int throws Exception*
9. *libcryptosec_crypto_asymmetric_key_string_get_algorithm($privateKey, $passwd = null) : string throws Exception*
10. *libcryptosec_crypto_get_public_key_stream($privateKey) : array throws Exception*
11. *libcryptosec_crypto_asymmetric_key_stream_get_size($privateKey) : int throws Exception*
12. *libcryptosec_crypto_asymmetric_key_stream_get_algorithm($privateKey) : string throws Exception*
13. *libcryptosec_crypto_generate_private_key($algorithm, $size, $cipher, $passwd= null) : string throws Exception*
14. *libcryptosec_crypto_encrypt_private_key($privateKey, $passwd, $cipher = "DES_EDE3") : string throws Exception*
15. *libcryptosec_crypto_generate_symmetric_key($algorithm, $size) : stream*
16. *libcryptosec_crypto_symmetric_encrypt_stream($key, $data, $algorithm, $operationMode) : stream throws Exception*
17. *libcryptosec_crypto_symmetric_encrypt_string($key, $data, $algorithm, $operationMode) : stream throws Exception*
18. *libcryptosec_crypto_symmetric_decrypt($key, $data, $algorithm, $operationMode) : stream throws Exception*
19. *libcryptosec_crypto_assymmetric_encrypt_stream($publicKey, $data) : stream throws Exception*
20. *libcryptosec_crypto_assymmetric_encrypt_string($publicKey, $data) : stream throws Exception*
21. *libcryptosec_crypto_assymmetric_decrypt($data, $privateKey, $password = null) : stream throws Exception*
22. *libcryptosec_crypto_assymmetric_decrypt_with_engine($data, $enginePath, $engineID, $keyName, $commands = NULL) : stream throws Exception*

23. *libcryptosec_crypto_get_hash_stream($data, $algorithm) : stream*

24. *libcryptosec_crypto_get_hash_string($data, $algorithm) : stream*

3.3 TECNOLOGIAS UTILIZADAS

Para a implementação da Modcryptosec foi utilizado o Zend Engine Extension, o qual permite criar uma extensão para PHP. Com esta tecnologia, foi possível utilizar uma biblioteca de funções implementada em C++ (a libcryptosec) e fazer as conversões necessárias para fornecer as funções necessárias para PHP. Maiores informações sobre como criar uma extensão para PHP podem ser encontradas em (EXTENSION, 2007).

Com esta tecnologia é possível realizar chamadas à funções em PHP, passar parâmetros na chamada à função, e receber o valor de retorno em PHP. No item 3.3.1 será exemplificado o uso da Extensão. Os seguintes tipos de parâmetros são suportados:

- *Boolean*
- *Long*
- *Double*
- *String*
- *Resource*
- *Array*
- *Object*
- *Zval*

Para a implementação da Modcryptosec foram utilizados os tipos boolean, long, double, string, resource, array e zval. O tipo Object permite a utilização de objetos, no entanto, a complexidade envolvida para fazer a conversão de objetos em PHP para C++ e vice-versa acabaram fazendo a idéia de trabalhar com objetos na extensão ser abandonada. Seria necessário alocar manualmente memória para cada atributo, cada método e cada estrutura de dados presente no objeto, ainda refazendo o mesmo processo para os objetos associados àquele. O tempo de

Tabela 1: Comparação de funções de manipulação de arrays em PHP e C++.

Em PHP	**Em C++**
$arr = array();	array_init(arr);
$arr[] = NULL;	add_next_index_null(arr);
$arr[] = 42;	add_next_index_long(arr, 42);
$arr[] = true;	add_next_index_bool(arr, 1);
$arr[] = 3.14;	add_next_index_double(arr, 3.14);
$arr[] = 'foo';	add_next_index_string(arr, "foo", 1);
$arr[] = $myvar;	add_next_index_zval(arr, myvar);
$arr[0] = NULL;	add_index_null(arr, 0);
$arr[1] = 42;	add_index_long(arr, 1, 42);
$arr[2] = true;	add_index_bool(arr, 2, 1);
$arr[3] = 3.14;	add_index_double(arr, 3, 3.14);
$arr[4] = 'foo';	add_index_string(arr, 4, "foo", 1);
$arr[5] = $myvar;	add_index_zval(arr, 5, myvar);
$arr['abc'] = NULL;	add_assoc_null(arr, "abc");
$arr['def'] = 711;	add_assoc_long(arr, "def", 711);
$arr['ghi'] = true;	add_assoc_bool(arr, "ghi", 1);
$arr['jkl'] = 1.44;	add_assoc_double(arr, "jkl", 1.44);
$arr['mno'] = 'baz';	add_assoc_string(arr, "mno", "baz", 1);
$arr['pqr'] = $myvar;	add_assoc_zval(arr, "pqr", myvar);

aprendizado para trabalhar com objetos seria bastante grande, e o prazo de desenvolvimento do projeto estava curto.

Um recurso bastante interessante do Zend são suas funções de manipulação de arrays. O Zend Engine torna a troca de arrays complexos entre as duas linguagens muito simples, provendo uma série de funções para utilização em C++ onde se é possível manipular os arrays extamente do mesmo modo que na linguagem de programação PHP:

Desta forma, quando se faz necessário a utilização de arrays na linguagem de programação C++, o mesmo pode ser montado utilizando as funções do Zend, não requerindo a conversão de formatos para a utilização em PHP. O array com esta estrutura pode ser retornado diretamente para o PHP como valor de retorno da função que foi chamada.

Outro recurso utilizado bastante importante nessa implementação foi a Application Programming Interface (API) de streams do Zend. Ver (STREAMS..., 2007). Em diversos casos, é necessário passar como parâmetro de função, ou como retorno de função uma string no formato binário. O problema é que na conversão da string de uma linguagem para outra, o caractere nulo, no formato American Standard Code for Information Interchange (ASCII), da string é reconhecido como final da string. Isso torna inviável a passagem de textos binários via string entre

as linguagens. A utilização de streams soluciona o problema levantado. Uma das linguagens abre o stream, escreve os dados binários, e passa o stream como parâmetro ou como retorno da função à outra linguagem, que por sua vez lê os dados contidos no stream sem perda de informação. Após a operação de leitura, o stream deve ser fechado.

3.3.1 ESCREVENDO UM MÓDULO PARA PHP

Nesta seção será mostrado através de um exemplo simples como desenvolver um módulo para PHP. O exemplo utilizado consiste no desenvolvimento de uma das funções implementada na Modcryptosec. Será implementada a função "*libcryptosec_certificate_get_public_key($certData) : string*" e serão utilizados os mesmos valores das definições da própria Modcryptosec.

O exemplo foi implementado sob o sistema operacional Linux Ubuntu v7.04. Os requisitos necessários para a implementação do módulo são:

- compilador *phpize*
- pacote *php5* e *php-ext*

Primeiramente, deve ser criada a estrutura básica necessária para possibilitar o uso das funções do Zend Engine na linguagem de programação C++. Os seguintes arquivos são necessários:

- **config.m4**: Arquivo de configuração do compilador phpize. É utilizado pelo compilador para especificar as características da compilação e do arquivo de biblioteca ("modcryptosec.so") a ser gerado;
- **php.h**: Especificação das funções do Zend Engine, contendo a definição das funções do Zend Engine a serem utilizadas na implementação do módulo;
- **modcryptosec.h**: Cabeçalho onde o usuário define as funções a serem implementadas no módulo;
- **modcryptosec.cpp**: Código-fonte do módulo que implementará as funções especificadas no arquivo de cabeçalho (modcryptosec.h). Deverá incluir o arquivo ("php.h").

A Figura 6 mostra como foi implementado o arquivo de configuração config.m4. A Figura 7 mostra a estrutura do arquivo de cabeçalho modcryptosec.h e a Figura 8 exibe a implementação

```
1  PHP_ARG_ENABLE(modcryptosec, whether to enable modcryptosec support,
2  [ --enable-modcryptosec   Enable modcryptosec support])
3
4  if test "$PHP_MODCRYPTOSEC" = "yes"; then
5    AC_DEFINE(HAVE_MODCRYPTOSEC, 1, [Whether you have modcryptosec])
6    PHP_NEW_EXTENSION(modcryptosec, modcryptosec.cpp, $ext_shared)
7  fi
8  $AA = "aa"
9
10 PHP_ADD_INCLUDE(/usr/include/modcryptosec)
11 PHP_ADD_LIBRARY(cryptosec,1, MODCRYPTOSEC_LIBADD)
12 #PHP_ADD_LIBPATH(/usr/include/modcryptosec, MODCRYPTOSEC_LIBADD)
13
14 CC=g++
15 PHP_SUBST(CC)
16
17 PHP_SUBST(MODCRYPTOSEC_LIBADD)
```

Figura 6: Conteúdo do arquivo config.m4.

```
1  #ifndef PHP_MODCRYPTOSEC_H
2  #define PHP_MODCRYPTOSEC_H 1
3
4  #ifdef ZTS
5  #include "TSRM.h"
6  #endif
7
8  #include "zend_exceptions.h"
9
10 #ifdef ZTS
11 #define MODCRYPTOSEC_G(v) TSRMG(modcryptosec_globals_id,
12         zend_modcryptosec_globals *, v)
13 #else
14 #define MODCRYPTOSEC_G(v) (modcryptosec_globals.v)
15 #endif
16
17 //especificar as funções aqui
18 PHP_FUNCTION(libcryptosec_certificate_get_public_key);
19
20 extern zend_module_entry modcryptosec_module_entry;
21 #define phpext_modcryptosec_ptr &modcryptosec_module_entry
22 #endif
```

Figura 7: Conteúdo do arquivo modcryptosec.h.

das funções do módulo. Para alterar o nome do módulo, basta substituir a palavra Modcryptosec pelo novo nome em todos os três arquivos.

Uma vez que essa estrutura esteja montada, com todas as funções do módulo especificadas e devidamente implementadas e com o arquivo de configuração corretamente definido, podemos compilar o código e gerar o arquivo de biblioteca, neste caso definido no config.m4 como modcryptosec.so. Para tal, pode-se utilizar um terminal qualquer e entrar com os comandos exibidos na Figura 9. Nas linhas 3, 4 e 5 o código é compilado e o arquivo modcryptosec.so é gerado. Na linha 9, o arquivo modcryptosec.so é copiado para o diretório de módulos do php. Para utilizar as funções do módulo basta especificar incluir o módulo no código php, através da função *dl(modcryptosec.so)*, que irá abrir seu diretório padrão de módulos e carregar a modcryptosec.

A partir deste ponto, as funções do módulo podem ser utilizadas em PHP. A Figura 10

```
#ifdef HAVE_CONFIG_H
#include "config.h"
#endif

//includes necessários ao módulo
#include "php.h"
#include "modcryptosec.h"
//includes da libcryptosec
#include <libcryptosec/certificate/Certificate.h>
#include <libcryptosec/KeyPair.h>
#include <libcryptosec/exception/CertificationException.h>
//demais includes
#include <string.h>
#include <stdlib.h>

static function_entry modcryptosec_functions[] = {

    //Especificar aqui as funções pertencentes ao módulo
    PHP_FE(libcryptosec_certificate_get_public_key, NULL)

    {NULL, NULL, NULL}
};

PHP_FUNCTION(libcryptosec_certificate_get_public_key)
{
        char *strcert;
        int strcert_len;

        if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "s",
                        &strcert, &strcert_len) == FAILURE){
                RETURN_NULL();
        }

        string sData(strcert);
        Certificate* cert = new Certificate(sData);
        PublicKey *pubkey = cert->getPublicKey();
        string retorno = pubkey->getPemEncoded();

        delete cert;
        delete pubkey;

        char pkeyretorno[8192];
        strcpy(pkeyretorno, retorno.c_str());
        RETURN_STRING(pkeyretorno, 1);
}

```

Figura 8: Conteúdo do arquivo modcryptosec.cpp.

mostra um exemplo de utilização da função do módulo implementada. A Figura 11 mostra o resultado da execução do código php.

```
Compila o código e gera o arquivo de biblioteca

        $ phpize
        $ ./configure --enable-modcryptosec
        $ make

Copia o arquivo gerado para o diretório de módulos do php

        # cp modules/hello.so /usr/lib/php5/20060613+lfs/
```

Figura 9: Sequência de comandos para compilar o módulo.

```
<?php

dl("modcryptosec.so");

$certificado = file_get_contents("AcIntermediaria.pem");
$publicKey = libcryptosec_certificate_get_public_key($file);
var_dump($publicKey);

?>
```

Figura 10: Exemplo de utilização da Modcryptosec.

```
array(3) {
  ["key"]=>
  string(451) "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAw0K8cY/SoS7tIEuMBxgR
cqcEMEb4+sf1SwQMwuVVPx4bl4BOl93UvPLW8KpueFHv1efD4ovDE4epUnQVL3vi
JuYB7LNVOMirbln1PU61gyrLmTNnKsbIlVbJxIqzrBOaDKLnen6Ghq1T2Tj5nuPy
EgNGlJFIB6OLL/6iUjB3Y3KMsZyX27PDw5JDMaOXYG1Cedlk8svwtEVbVoWCUg3i
FU8RX/bsGXF5v4WHoD3O35WF2enSQlXpvpyIa8biLiJPBvFXGBJ/JLLhU4EQk2t9
l8K68OTNbdDbOQx+5nK/v1vCJReLXvF74VuzdMVdK7bZwmQI7oe+qmnmg5YEDAQt
HwIDAQAB
-----END PUBLIC KEY-----
"
  ["size"]=>
  int(2048)
  ["algorithm"]=>
  string(3) "RSA"
}
```

Figura 11: Exemplo de utilização da Modcryptosec.

4 PROJETO DA AUTORIDADE CERTIFICADORA TEMPORAL

Uma ACT é uma entidade pertencente a uma ICPT, tendo como função prover temporalidade aos documentos eletrônicos. Nesta seção será tratado o projeto desenvolvido de ACT, abordando todos os serviços disponibilizados por esta autoridade, bem como a relação da ACT com as demais autoridades de uma ICPT.

O funcionamento de uma ACT se baseia na criação e manipulação de um par de chaves correspondentes, utilizando-se da criptografia assimétrica. A uma destas chaves se dá o nome de chave de sigilo (ou chave pública) e a outra, de chave de liberação (ou chave privada). A chave de sigilo é publicada através de um certificado digital no formato X.509, contendo esta chave e a assinatura digital da ACT. A ACT se responsabiliza em manter sob sigilo a chave de liberação até a data de liberação do certificado temporal ser atingida. Neste momento, a ACT publica esta chave (de liberação), a qual pode ser utilizada pelos usuários para decifrar os documentos cifrados com a chave de sigilo publicada anteriormente. A Figura 12 detalha o processo de funcionamento da ACT.

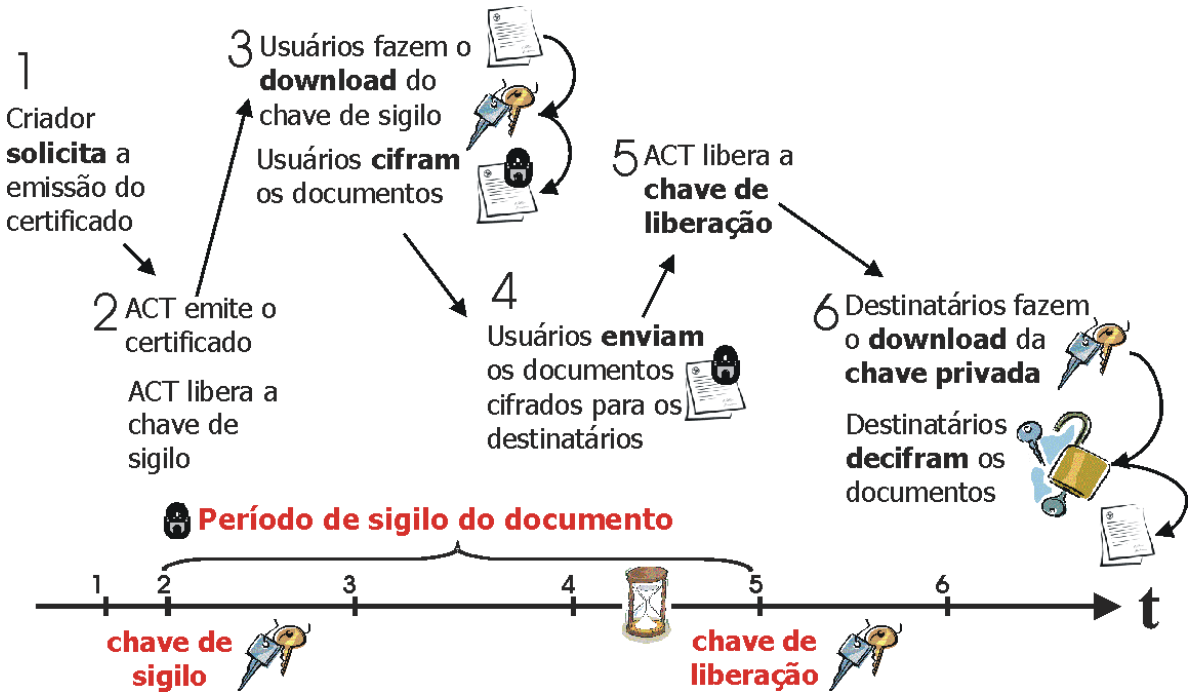

Figura 12: Funcionamento de uma Infra-estrutura em Chaves Públicas Temporal.

Como podemos ver na Figura 12, o funcionamento da ACT consiste, basicamente, nos seguintes passos:

1. O usuário solicita a criação do certificado temporal;
2. A ACT gera as chaves de sigilo e de liberação, publica o certificado X.509 contendo a chave de sigilo e armazena de forma segura a chave de liberação;
3. Os usuários interessados fazem o download do certificado temporal e utilizam a chave de sigilo contida nele para cifrar seus documentos;
4. Os usuários enviam seu(s) documento(s) cifrado(s) para o(s) destinatário(s) desejado(s);
5. Uma vez atingida a data de liberação do certificado temporal, a ACT publica a chave de liberação referente à chave de sigilo publicada anteriormente, no passo 2;
6. Finalizando o processo, o(s) destinatário(s) pode(m) fazer o download da chave de liberação e decifrar o(s) documento(s) recebidos.

4.1 CERTIFICADOS TEMPORAIS

O certificado temporal emitido pela ACT trata-se de um certificado digital no padrão X.509v3. O paddrão X.509 foi criado e aprovado pela ITU-T, o qual também foi aceito pela IETF, que descreve sua versão na RFC3280. Maiores informações sobre o X.509 podem ser encontradas em (FORD; BAUM, 2000).

A Tabela 2 mostra os campos de um certificado digital X.509v3 e suas descrições. O certificado temporal da ACT será composto por todos os campos dessa tabela, sendo que a extensão UserNotes será adicionada ao certificado, onde será especificada a data de liberação do certificado temporal.

Os campos Subject e Issuer armazenam as informações do criador do certificado e do certificado que o assinou. A Tabela 3 mostra as principais informações armazenados relevantes à ACT.

Através da obtenção da chave pública do certificado digital, os usuários poderão utilizá-la para cifrar os documentos necessários. No caso de um certificado temporal esta chave pública é conhecida como chave de sigilo, uma vez que eles podem manter seus documentos sob sigilo através dela.

Tabela 2: Campos de um certificado digital X.509v3

Campo	**Descrição**
Version	A versão do certificado X.509
Serial Number	Um dos campos utilizados para identificar o certificado de forma única
Signatura Algorithm	Algoritmo utilizado para assinar o certificado
Issuer	Identifica o assinador do certificado
Validty	Especifica a data de criação (NotBefore) e a data de validade (Not After) do certificado
Subject	Identifica o criador do certificado
Subject Public Key Info	Contém a chave pública e suas informações relevantes (e.g. algoritmo utilizado como RSA ou DSA, tamanho da chave).
Extensions	Compreende as informações adicionais do certificado, através do uso de diversas extensões
Signature	A assinatura do certificado, assinada por uma determinada chave privada

Tabela 3: Informações sobre o assinador ou criador do certificado

Campo	**Descrição**
Common Name (CN)	Nome comum. Armazena o nome da entidade
Country (C)	País ao qual a entidade pertence
Organization (O)	Nome da organização ao qual a entidade pertence
Organization Unit (OU)	Unidade da organização ao qual a entidade pertence
State or Province (SP)	Estado ou província ao qual a entidade pertence
Serial Number (S)	O número serial do certificado do assinador
Locality (L)	Localidade ao qual pertence a entidade. Geralmente utiliza-se a cidade
Email (E)	E-mail da entidade

No projeto da Autoridade Certificadora Temoral existirão dois tipos de certificados temporais. Ambos possuem a mesma estrutura, mas serão utilizados para finalidades diferentes:

- **Certificados temporais de propósito geral**: São os certificados emitidos pelos administradores da ACT, onde qualquer usuário pode utilizar os certificados. Seu propósito é permitir a criação de certificados temporais genéricos que possam ser utilizados para várias aplicabilidades diferentes. Eventualmente, o administrador da ACT pode criar um certificado temporal com data de liberação do mesmo para cada dia do ano, atendendo a necessidade da maioria dos usuários da ACT.
- **Certificados temporais com políticas especiais**: São emitidos pelos usuários da ACT. Caso o usuário não encontre um certificado temporal de propósito geral que atenda às suas necessidades, pode solicitar junto a ACT um certificado temporal com políticas especiais para o seu caso.

O certificado temporal é assinado pela ACT, a qual atesta a validade do certificado e se compromete em manter a chave privada (ou chave de liberação) sob sigilo até o momento definido.

4.2 MÓDULOS DA AUTORIDADE CERTIFICADORA TEMPORAL

A ACT será utilizada por cinco diferentes classes de usuários: criador, administradores, usuários cadastrados, usuários geral e auditores. Basicamente, o criador define e cria a estrutura inicial da ACT, a qual fica sob responsabilidade dos administradores cadastrados nesse processo de criação. Os administradores gerenciam o uso da ACT, fornecendo os serviços para os usuários da autoridade. Os auditores realizam as operações de auditoria das atividades da ACT, fiscalizando o trabalho dos administradores. E os usuários, por sua vez, apenas utilizam os serviços fornecidos pela autoridade. Nesse contexto, para estruturar o projeto da ACT, ela foi dividida nos seguintes módulos:

- **Módulo Criador:** responsável pela criação da ACT e pela definição das configurações iniciais;
- **Módulo Administrador:** responsável por fornecer os serviços aos administradores da ACT;
- **Módulo Usuário:** responsável por fornecer os serviços aos usuários cadastrados na ACT;

- **Módulo Público:** responsável pelas funções públicas da ACT;
- **Módulo Auditor:** resposável pela auditoria das operações efetuadas pela ACT;

Os criadores, administradores, usuários cadastrados e auditores deverão estar devidamente cadastrados no sistema. Apenas os usuários do Módulo Público não necessitam ter seu cadastro. Para ter acesso ao módulo correspondente, os usuários deverão realizar a operação de login, ora fornecendo sua senha, ora fornecendo seu certificado digital. Os dois modos de login deverão ser suportados pelo sistema: via senha ou via certificado digital.

Nas próximas seções será realizada uma abordagem detalhada sobre cada um dos módulos que irão compôr a ACT e suas respectivas funcionalidades.

4.2.1 MÓDULO CRIADOR

O Módulo Criador é o módulo responsável pela criação, propriamente dita, da ACT. O usuário criador prepara a autoridade para os administradores, que a gerenciam. Este módulo provê as seguintes funcionalidades ao criador da ACT:

1. Cadastrar uma ACT;
2. Cadastrar/Editar/Remover administradores;
3. Cadastrar/Editar/Remover CSMs (Módulo de Relógio Seguro)

O criador gera uma instância da ACT a ser criada e cadastra os administradores na instância gerada. O criador é o único usuário da ACT que tem o poder de alterar a equipe que irá gerenciar a autoridade. Os CSMs foram definidos no Item 2.2.3 deste trabalho, e seu cadastro, alteração ou não utilização também fica a cargo do criador.

O processo de criação da ACT ainda depende de algumas atividades dos admnistradores do sistema. Neste momento, a ACT ainda não pode gerar os certificados temporais aos usuários.

4.2.2 MÓDULO ADMINISTRADOR

Este módulo é responsável por fornecer aos administradores do sistema, todos os serviços do nível de gerenciamento da ACT. O Módulo Administrador permite aos administradores:

1. Cadastrar ou alterar o certificado digital da ACT;

2. Solicitar a revogação do certificado digital da ACT;
3. Definir as políticas da ACT;
4. Aprovar ou recusar solicitações de cadastros de usuários;
5. Definir as políticas de uso para cada usuário cadastrado na ACT;
6. Emitir os certificados temporais de propósito geral da ACT;
7. Revogar tanto os certificados temporais de propósito geral quanto os de políticas especiais;
8. Cadastrar/Editar/Remover templates de políticas para os usuários ou para ACT.
9. Editar dados pessoais;

Primeiramente, para uma ACT poder iniciar suas atividades, necessita-se de um certificado digital assinado por uma AC oficial, o que irá tornar a ACT um membro de uma determinada cadeia de confiança. No processo de criação, os administradores deverão cadastrar o certificado digital da ACT. No caso do comprometimento da chave privada da ACT os administradores devem gerar uma requisição de revogação do certificado digital e submetê-la a AC que emitiu o certificado. Após o processo de revogação, deverão cadastrar o novo certificado digital da autoridade, devidamente assinado por uma AC.

Em seguida, deve ser definida a política de uso da ACT. Somente a partir deste passo, a ACT pode iniciar suas atividades. No entanto, os administradores podem alterar a política da ACT, a qualquer momento que se fizer necessário. Qualquer administrador cadastrado tem este poder, uma vez que lhe foi atribuído (pelo criador) a confiança de gerenciar a ACT. A política de uso da ACT, deverá disponibilizar as opções de:

- Emitir a LCR de forma Manual ou Automática (definindo a periodicidade neste caso);
- Tempo adicional de validade do certificado;
- Definir local padrão para publição de certificados;
- Definir local para publicação de LCR;
- Configurações do par de chaves a ser gerado para os certificados temporais de propósito geral (e.g. tamnho da chave, algoritmo, tipo);
- Algoritmo utilizado na assinatura digital dos certificados temporais emitidos pela ACT;

A política dos usuários da ACT, deverá disponibilizar as opções de:

- Definir o número máximo de certificados que um usuário pode emitir em um determinado período de tempo;
- Período de tempo válido para o cadastro do usuário;
- Configurações do par de chaves a ser gerado para os certificados temporais (com políticas específicas) emitidos pelo usuário (e.g. tamnho da chave, algoritmo, tipo);
- Algoritmo de assinatura dos certificados temporais do usuário;

Os administradores podem criar os templates de políticas tanto à ACT, quanto aos usuários. Uma configuração específica de política é um template, o que facilita o processo de gerenciamento de políticas. Por exemplo, a ACT pode criar templates para vários perfis de usuários ou poderá criar templates para o uso da ACT mais seguros, ou mais econômicos.

Todos os usuários que venham a efetuar seu cadastro junto a ACT, deverão ter seu cadastro aprovado por um administrador. No processo de aprovação do cadastro do usuário, o administrador deverá definir a política de uso para o usuário, selecionando um template pré-cadastrado. Usuários diferentes podem ter permissões diferentes de utilização dos serviços da ACT. Se o cadastro for um serviço pago, é facilmente justificável a existência de diferentes perfis de usuários na ACT.

Os administradores são responsáveis por emitir os certificados de propósito geral. Em uma ACT com uma utilização em massa, seus administradores poderiam, por exemplo, emitir um certificado temporal com uma data de liberação para cada dia do ano, evitando a emissão de certificados temporais em excesso pelos usuários A maioria das necessidades dos usuários deveriam ser atendidas, exceto os casos em que a a chave de liberação necessite ser liberada em um horário bem específico durante o dia

Os administradores também podem alterar seus dados pessoais, previamente cadastros pelo criador do sistema, alterando sua senha ou seu certificado temporal que o identificam no sistema.

4.2.3 MÓDULO USUÁRIO

O Módulo Usuário compreende os usuários devidamente cadastrados na ACT. Para tal, eles necessitam realizar uma solicitação de cadastro à ACT, o qual será aprovado ou não pelos administradores. Estes usuários não apenas utilizam os certificados digitais temporais gerados pela

ACT, ou por algum outro usuário da ACT como também geram seus próprios certificados respeitando suas políticas de usuário (definidas no item 4.2.2). Este módulo permite aos usuários cadastrados na ACT:

1. Emitir os certificados temporais com políticas especiais;
2. Prorrogar a data de liberação de seus certificados temporais emitidos;
3. Se necessário, solicitar a revogação de seus certificados temporais emitidos.

Se por algum motivo for necessário revogar o certificado temporal, o usuário pode solicitar a revogação do mesmo. A solicitação de revogação de um certificado temporal pelo usuário é automaticamente aprovada. O gerenciamento do certificado temporal emitido pelo usuário é de responsabilidade dele. E no caso de comprometimento da chave privada da ACT, os próprios administradores da ACT deverão revogar o certificado temporal.

4.2.4 MÓDULO PÚBLICO

Este módulo pode ser acessado por qualquer usuário, não sendo necessário um cadastro na ACT e consequentemente não necessitando realizar o processo de login. Os usuários deste módulo não tem permissão para gerar certificados temporais, mas podem utilizá-los, tantos os certificados temporais de propósito geral quanto os com políticas especiais. O Módulo Público permite, então, a qualquer usuário:

- Localizar qualquer um dos certificados temporais da ACT, tanto os de propósito geral quanto os com políticas especiais;
- Fazer o download da chave de sigilo e da chave de liberação (se for o caso) dos certificados temporais desejados e utilizá-las para seus devidos fins;
- Solicitar seu cadastro junto à ACT.

Se o usuário necessitar gerar seu próprio certificado temporal, uma vez que não exista nenhum certificado temporal que atenda suas necessidades, o mesmo pode solicitar seu cadastro junto à ACT. Esse cadastro será submedito à aprovação de um administrador da ACT.

4.2.5 MÓDULO AUDITOR

Os auditores do sistema pode verificar as operações realizadas pela ACT através dos logs e relatórios gerados por ela. Através da análise dos relatórios e logs é possível levantar irregularidades e operações ilegais que venham a estar ocorrendo com a utilização ou administração da ACT. Todos os eventos que ocorrem na ACT são registrados e podem posteriormente vistoriados.

4.3 INFRA-ESTRUTURA EM CHAVES PÚBLICAS TEMPORAL

Um exemplo de uma possível arquitetura de infra-estrutura em chaves públicas temporal pode ser visualizado na Figura 13. Existem três tipos de autoridades certificadoras. A AC-raíz, que possui seu certificado digital auto-assinado, as ACs regionais ou intermediárias (caso da AC-UFSC e da AC-UFPR da Figura 13), que têm permissão de assinar os certificados digitais de outras autoridades e as autoridades finais (AC LabSEC) que podem somente assinar certificados digitais de usuários. Cada autoridade certificadora pode ou não utilizar os serviços de um HSM. No exemplo, a AC-Raiz assinou digitalmente os certificados da AC-UFSC e da AC-UFPR. A AC-UFSC, por sua vez, assinou digitalmente os certificados da AC-LabSEC, da ACT-UFSC, e do servidor de carimbo de tempo da UFSC (SCT-UFSC). A SCT-UFSC fornece os serviços de data e hora à AC-LabSEC e à ACT-UFSC. Várias arquiteturas podem ser formadas, e neste exemplo a SCT-UFSC e a ACT-UFSC compartilham o mesmo dispositivo de hardware, o CSM. A ACT-UFSC é gerenciada por administradores, que por sua vez são fiscalizados por auditores. Os usuários dessa infra-estrutura comunicam-se com a ACT-UFSC através de uma interface WEB que provê os serviços de diretório público, cadastro de usuários e geração de requisições de criação de certificados temporais.

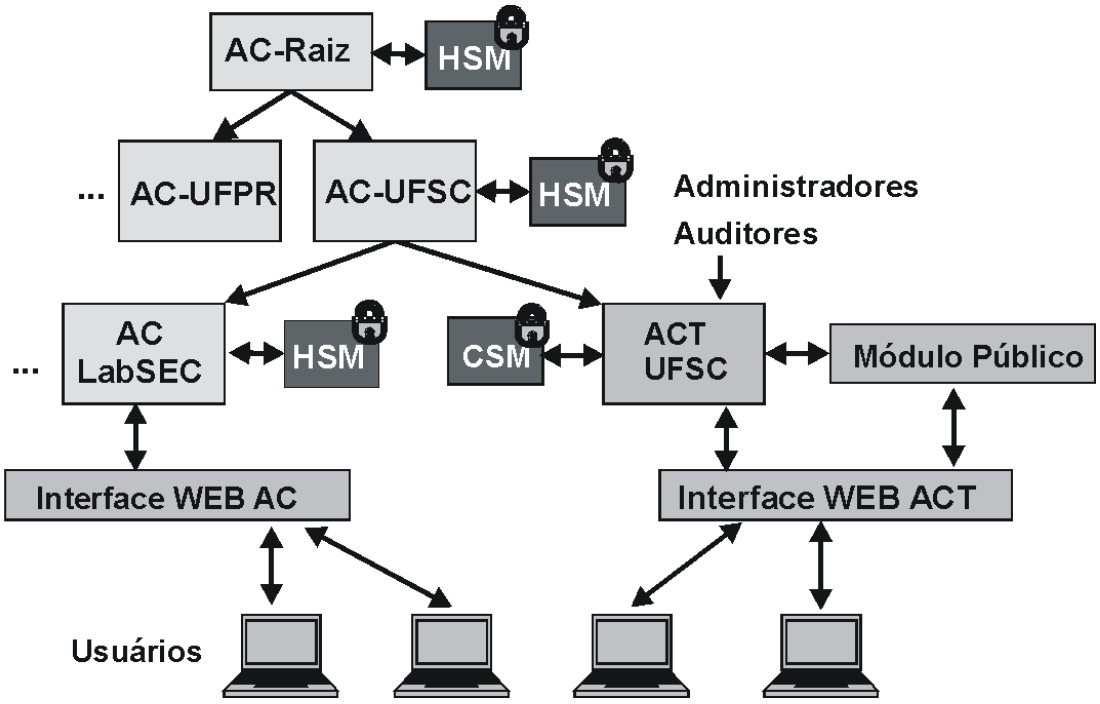

Figura 13: Exemplo de uma possível arquitetura de Infra-estrutura em Chaves Públicas Temporal

5 PROTÓTIPO DA AUTORIDADE CERTIFICADORA TEMPORAL IMPLEMENTADO

Inicialmente, a idéia do projeto era a implementação completa do projeto da Autoridade Certificadora Temporal desenvolvido. No entanto, como o desenvolvimento da libcryptosec acabou atendendo aos requisitos não somente da ACT mas também dos demais projetos do LabSEC, sua implementação acabou consumindo grande parte do tempo de desenvolvimento deste trabalho.

Como citado no Capítulo 3, da Modcryptosec, a linguagem de programação escolhida para a implementação foi o PHP, pela facilidade de desenvolvimento de interfaces WEB, e também pela facilidade de aprendizado da linguagem de programação no ambiente do LabSEC, uma vez que várias pessoas tinha conhecimento da linguagem e diversos outros projetos do laboratório estavam sendo desenvolvidos nesta linguagem. Isso facilita o desenvolvimento e mesmo a futura manutenção da implementação. O protótipo utilizou o módulo de funções implementado no Capítulo 3. O módulo fornece todas as funções criptográficas necessárias à implementação da ACT, e o protótipo implementado, disponibiliza a interface gráfica de acesso à ACT, e realiza as operações de interfaceamento, controle, atualização do banco de dados e utiliza as funções criptográficas.

Dentre os Módulos projetados no Capítulo 4, o Módulo Público e o Módulo Usuário foram implementados, e o Módulo Administrador, parcialmente. As funcionalidades disponibilizadas pelo sistema compreendem o cadastro de usuários no sistema, a emissão e utilização dos certificados temporais da ACT. As funções de criação da ACT são realizadas manualmente e algumas funções de gerenciamento não foram implementadas, não havendo suporte para algumas operações ou não havendo interface gráfica para outras.

O protótipo desenvolvido não implementa a estrutura completa definida no projeto da ACT, mas possui a estrutura básica para permitir a emissão de certificados digitais temporais, respeitando os requisitos de segurança de sigilo, autenticação, não-repúdio, integridade e temporalidade. Neste capítulo, o protótipo da Autoridade Certificadora Temporal será descrito em detalhes, abordadando as funcionalidades do sistema e as tecnologias utilizadas para o desenvolvimento do protótipo.

5.1 DESCRIÇÃO DO PROTÓTIPO

O acesso à ACT se realiza via WEB. Ao acessar o endereço da ACT, o usuário será redirecionado para a página de login (Figura 15). Nesta página, o usuário poderá efetuar o login como usuário do sistema, efetuar o login como administrador do sistema, utilizar simplesmente as funções públicas da ACT, ou então solicitar seu cadastro. A Figura 14 mostra o mapa do site, mostrando a estrutura em que os serviços da ACT são disponibilizados.

```
Tela de Login
        Opções do usuário (Modulo Usuário)
                Emitir certificado temporal
                        Confirmação da emissão
                Certificados temporais do usuário
                        Confirmação de prorrogação
                        Confirmação de revogação
                Certificados temporais do Módulo Publico
                Alterar dados pessoais
                        Confirmação da alteração
        Opções do administrador (Módulo Administrador)
                Solicitar novo certificado temporal
                        Confirmação da emissão
                Gerenciar certificados temporais da ACT
                        Confirmação de prorrogação
                        Confirmação de revogação
                Consultar usuários da ACT
        Cadastro de usuário (Módulo Publico)
        Certificados temporais da ACT (Módulo Público)
```

Figura 14: Caminhos de navegação do protótipo da ACT implementado.

Figura 15: Página inicial da ACT. Opções de login, cadastro de usuário e consulta aos certificados temporais da autoridade disponíveis.

Os administradores e usuários cadastrados na ACT possuem seu cadastro na ACT, e reali-

zam o login através de seu nome de usuário e sua senha. Os administradores serão direcionados ao Módulo Administrador e os usuários, ao Módulo Usuário. Se o usuário não possuir um cadastro, poderá consultar os certificados temporais da ACT, através do link "Certificados Temporais" da página.

5.1.1 MÓDULO PÚBLICO

CADASTRO DE USUÁRIO

O usuário que desejar realizar o seu cadastro junto à ACT, deverá clicar no link "aqui" pertencente ao texto "Não está cadastrado? Clique aqui e faça seu cadastro".

Em seguida, o mesmo deverá preencher a ficha de cadastro e clicar no botão "Solicitar Cadastro", como podemos ver na Figura 16.

Figura 16: Tela de cadastro de usuário.

CONSULTA DOS CERTIFICADOS TEMPORAIS DA ACT

Para consultar os certificados temporais da ACT, o usuário deve clicar no link "Certificados Temporais" na tela de login, pertencente ao texto "Consulte os Certificados Temporais da ACT". Serão exibidos todos os certificados temporais da ACT, inclusive os certificados temporais emitidos por usuários. O usuário poderá visualizar os certificados, e fazer o download

do certificado digital temporal desejado ou da chave de liberação do certificado temporal se o tempo de liberação já tiver sido atingido. Os seguintes campos serão exibidos:

- **Nome Comum**: Corresponde ao campo Nome Comum do certificado temporal, e corresponde a entidade emissora do certificado.
- **Data de criação**: Corresponde a data e hora em que o certificado temporal foi emitido.
- **Data de liberação**: Corresponde a data e hora em que a chave de liberação deverá ser publicada na ACT.
- **Propósito**: Corresponde ao propósito ao qual foi criado o certificado temporal.
- **Download Certificado**: Esta opção sempre estará disponível, e possibilita a realização do *download* do certificado temporal. O mesmo pode ser utilizado para cifrar os documentos que necessitam permanecer em sigilo, até a data de liberação do certificado temporal.
- **Download da chave de liberação**: Quando a data de liberação do certificado temporal for atingida, será disponibilizada a opção de download da mesma. Caso a data atual corresponda a um período anterior a data data de liberação, será exibido o texto "Aguarde Liberação".

A tela de consulta dos certificados temporais do Módulo Público se assemelha muito à Figura 23, que consiste na tela de Administração dos Certificados Temporais. A única diferença é que as opções "Excluir" e "Prorrogar" não serão disponibilizadas uma vez que os usuários do Módulo Público não possuem esta permissão, para revogar ou prorrogar a data de liberação dos certificados temporais da ACT.

5.1.2 MÓDULO USUÁRIO

Uma vez que o usuário tenha efetuado seu login com seu nome de usuário e senha pré-cadastrados na opção de Cadastro de Usuário, descrito do Item 5.1.1 deste trabalho, estará no Módulo Usuário. Serão disponibilizadas os links para a solicitação de um certificado temporal com políticas especiais, gerenciamento dos certificados temporais emitidos por este usuário, consulta dos certificados temporais da ACT e alteração dos dados pessoais. As próximas seções descrevem as operações possíveis no Módulo Usuário.

- Solicitar Novo Certificado Temporal
- Gerenciar Certificados Temporais

Figura 17: Módulo Administrador, tela de gerenciamento dos certificados temporais da ACT.

- Consultar Certificados Temporais da ACT
- Alterar Dados Pessoais

SOLICITAR CERTIFICADO TEMPORAL

Para solicitar um novo certificado temporal, o usuário deverá selecionar a opção “Solicitar Novo Certificado Temporal” na tela principal do Módulo Usuário. Será exibido um formulário para o preenchimento dos dados para a emissão do certificado temporal. A Figura 21 mostra esta tela e os campos que devem ser preenchidos para realizar a solicitação. O usuário deve preencher os dados, informar a data de liberação do certificado temporal, seu propósito e as demais informações e então clicar no botão “Solicitar Certificado”.

Se todas as informações digitadas estiverem corretas, a ACT irá confirmar a emissão do certificado temporal, e exibirá um link para o download do mesmo. A Figura 19 mostra a tela de confirmação.

A partir deste momento, o usuário poderá prorrogar a data de liberação de seu certificado temporal, ou mesmo revogá-lo se for necessário.

GERENCIAR CERTIFICADOS TEMPORAIS

Esta tela permite a consulta aos certificados temporais emitidos por este usuário, e disponibiliza as opções de download do certificado temporal, download da chave de liberação (se for o caso), revogação e prorrogação da data de liberação do certificado temporal. Quando o usuário selecionar a opção “Gerenciar Meus Certificados Temporais” na tela principal do Módulo Usuário, a tela será mostrada. A Figura 20 mostra esta tela, exibindo um exemplo onde

Figura 18: Módulo Usuário, tela de solicitação de certificado temporal.

são exibidos dois certificados temporais emitidos pelo usuário João Silva, o primeiro com uma data de liberação futura e o segundo certificado temporal com a data de liberação atingida.

A partir desta tela o usuário pode prorrogar ou revogar seus certificados temporais.

PRORROGAR CERTIFICADO TEMPORAL

Uma vez que o usuário tenha selecionado a opção “Prorrogar” de seu certificado temporal, no Item 5.1.2 deste trabalho, uma tela exibirá a antiga data de liberação do certificado temporal e solicitará a nova data de liberação do certificado temporal. A Figura 22 mostra esta tela, onde a liberação do certificado temporal será prorrogada em um dia.

Figura 19: Módulo Usuário, tela de confirmação da emissão de certificado temporal.

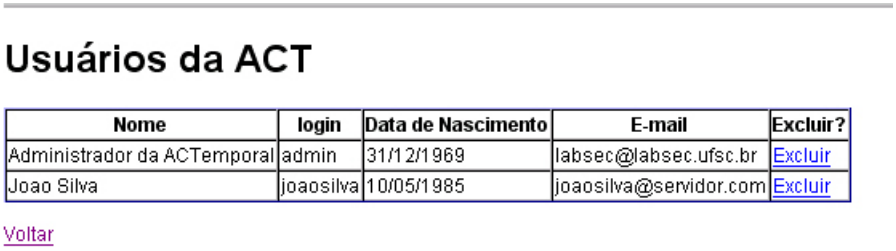

Figura 20: Módulo Usuário, tela de gerenciamento dos certificados temporais do usuário.

REVOGAR CERTIFICADO TEMPORAL

Se o usuário selecionar a opção “Excluir” do certificado temporal, será solicitada a confirmação da revogação e uma tela de confirmação da exclusão será exibida.

ALTERAR DADOS PESSOAIS

Para acessar esta opção, o usuário deve clicar na opção “Alterar Dados Pessoais” na tela principal do Módulo Usuário. Uma tela semelhante à da Figura 16 será exibida, onde o usuário poderá alterar seus dados convenientes e confirmar a alteração dos mesmos.

5.1.3 MÓDULO ADMINISTRADOR

Para ter acesso ao Módulo Administrador, o usuário deverá entrar com seu nome de usuário e senha na tela de login. Neste protótipo, os administradores são cadastrados diretamente no banco de dados, simulando o processo do usuário criador definido no projeto da ACT. O certificado digital da ACTemporal também é definido manualmente, através do upload do certificado

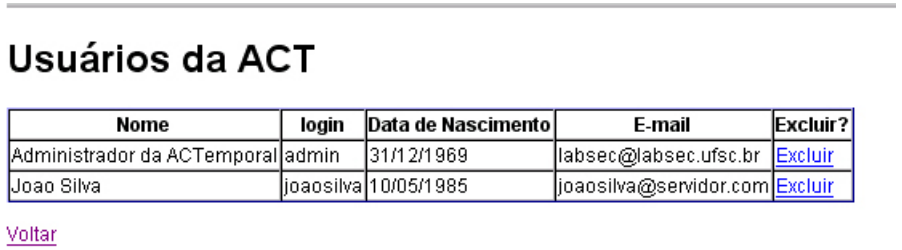

Figura 21: Módulo Usuário, tela de gerenciamento dos certificados temporais do usuário.

Figura 22: Módulo Usuário, tela de prorrogação de certificado temporal.

digital à estrutura da ACT. Esta estrutura de pastas utilizadas no projeto será exibida no Item 5.2.2 deste trabalho.

As opções de emissão de um novo certificado temporal, gerenciamento dos certificados temporais da ACT e consulta aos usuários cadastrados da ACT serão disponibilizadas ao administrador.

EMISSÃO DE UM CERTIFICADO TEMPORAL

Através desta opção, o administrador pode emitir os certificados temporais de propósito geral da ACT. O processo é idêntico ao descrito no item 5.1.2 deste capítulo.

GERENCIAMENTO DOS CERTIFICADOS TEMPORAIS

O administrador pode gerenciar todos os certificados temporais da ACT, tanto os emitidos pelos administradores da ACT, quanto os emitidos pelos usuários. São disponibilizadas

as opções de revogação e de prorrogação dos certificados temporais. A Figura 23 mostra esta tela. No exemplo, existem dois certificados temporais emitidos pelo usuário João Silva e dois certificados emitidos por administradores do sistema. Este processo se assemelha ao Item 5.1.2 do Módulo Usuário, exceto pelo fato de que os administradores tem privilégios de revogação e prorrogação não somente dos certificados emitidos por ele, mas por todos os certificados temporais da ACT. As operações de revogação e de prorrogação do certificado temporal são idênticas aos Itens 5.1.2 e 5.1.2 respectivamente do Módulo Usuário.

Figura 23: Módulo Administrador, tela de gerenciamento dos certificados temporais da ACT.

CONSULTAR USUÁRIOS DA ACT

O administrador tem permissão de consultar os usuários cadastrados na ACT. A Figura 24 mostra esta operação. O administrador pode remover os usuários cadastrados no sistema, apenas clicando no link "Excluir" no certificado temporal desejado. Uma solicitação de confirmação será exibida e em seguida, será confiramada a remoção.

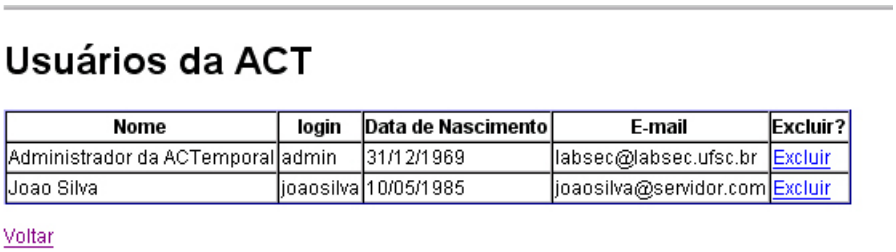

Figura 24: Módulo Administrador, tela de consulta aos usuários cadastrados na ACT.

5.2 TECNOLOGIAS UTILIZADAS NA IMPLEMENTAÇÃO

Nesta seção serão abordadas as tecnologias utilizadas para a implementação do protótipo da ACT. Um dos pontos levados em consideração durante a implementação foi a preocupação em utilizar softwares livres.

5.2.1 ECLIPSE PDT

O Eclipse é um conjunto de ferramentas de desenvolvimento *Open Source* que possibilita a extensão de novos recursos como a inclusão de frameworks. A versão utilizada foi a *Eclipse PHP Development Tools* (PDT), que disponibiliza uma infra-estrutura para o desenvolvimento em PHP. Outras informações sobre o Eclipse PDT podem ser conferidas em (PDT..., 2007).

5.2.2 ZEND FRAMEWORK

O Zend Framework é um *framework Open Source* (código aberto) que dispõe de vários recursos para desenvolvimento em PHP. O *framework* "propõe-se a oferecer uma biblioteca de recursos de grande poder, fornecendo soluções modernas, robustas e seguras para o desenvolvedor" (ZEND..., 2007).

O padrão de desenvolvimento do Zend Framework segue o padrão de projeto *Model-View-Controller* (MVC). Este padrão propõe a separação dos dados (Model) e da interface (View) criando uma camada entre eles para controlar (Controller) que processa os eventos da interface podendo alterar o modelo (MVC, 2007).

Uma facilidade importante para o projeto, é a abstração do banco de dados do Zend Framework. O framework utiliza o padrão de projeto adaptador (adapter) para interagir com o banco de dados. Com isso, caso for necessário trocar o banco de dados, basta apenas trocar as configurações do arquivo de configurações do banco de dados, dispensando a modificação das linhas de código. O componente responsável pela interação com o banco de dados é o **Zend_Db**. Para maiores informações consulte (ZEND_DB..., 2007).

O Zend Framework disponibiliza também várias funcionalidades importantes para aplicações web. Abaixo estão listadas alguns componentes que facilitam o desenvolvimento. Para maiores informações sobre os componentes consulte (PROGRAMMER'S..., 2007).

- **Zend_Acl**, controle de lista de acesso (access control list): controla os usuários e seus privilégios no sistema;

- **Zend_Auth**, controle de autenticação: responsável por autenticar os usuários no sistema;
- **Zend_Date**, manipulador de data: responsável por manipular datas e horários;
- **Zend_Pdf**, manipulador PDF: engine responsável pela manipulação de arquivos no formato Portable Document Format (PDF);
- **Zend_Mail**, manipulador de e-mails: responsável por compor e enviar e-mails;
- **Zend_Validate**, validador de campos: componente responsável pela validação de campos conferindo se respeitam os requisitos.

5.2.3 SVN

O *Subversion* (SVN) é um sistema de controle de versões que permite aos usuários verificar mudanças feitas em arquivos no formato digital. O SVN foi projetado para substituir o *Concurrent Version System* (CVS) [1] (SUBVERSION, 2007).

Com o SVN, é possível que várias pessoas trabalhem simultaneamente em arquivos do projeto sem comprometer uma o trabalho da outra, pois os arquivos ficam armazenados num servidor. Neste sistema, um *log* registra as modificações feitas no projeto mesclando as alterações feitas. Com isso, se houver alguma falha em alguma parte do projeto, basta retomar as versões anteriores dos arquivos.

5.2.4 POSTGRESQL

O PostgreSQL é um *software* gerenciador de banco de dados *Open Source* que implementa transações com Atomicidade, Consistência, Isolamento, Durabilidade (ACID) [2]. O *software* está disponível para vários sistemas operacionais (Windows, Unix e Linux) e inclui a maioria dos tipos especificados nos padrões SQL92 e SQL99 (ABOUT..., 2007).

O sistema possui uma ampla interface de programação, suportando as linguagens C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC entre outras (ABOUT..., 2007).

Este banco de dados foi escolhido para os testes locais, isto é, foi utilizado para os testes enquanto o CSM não estava disponível.

Devido ao padrão adaptar do Zend Framework, fica fácil transferir o protótipo da ACT para um CSM, pois exige apenas a mudança no arquivo de configurações do banco de dados.

[1] Para maiores informações consulte `http://pt.wikipedia.org/wiki/CVS`
[2] Para maiores informações consulte `http://pt.wikipedia.org/wiki/ACID`

5.3 PUBLICAÇÃO DO PROTÓTIPO

O projeto está disponível através do endereço `https://ac-temporal.labsec.ufsc.br`. Neste endereço é possível consultar e emitir certificados temporais atráves da Autoridade Certificadora Temporal da UFSC.

O site é protegido com criptografia SSL e o certificado é assinado pela AC-SSL (autoridade da UFSC responsável por assinar os certificados de sites da UFSC), que por sua vez possiu o certificado assinado pela AC-UFSC cujo certificado é assinado pela ICP-EDU. Com isso as informações sigilosas dos usuáriossão enviadas por um canal seguro.

6 CONSIDERAÇÕES FINAIS

O grande desafio da criptografia temporal é tornar possível a realização, em meio digital, dos serviços realizados sobre os documentos em papel que envolvam temporalidade. O protótipo implementado provê essa funcionalidade, embora ainda não disponha de todos os serviços que uma ACT deve fornecer. De qualquer forma, o objetivo principal deste trabalho de permitir o envio de documentos para o futuro de forma segura foi alcançado através da implementação do protótipo.

O desenvolvimento da Modcryptosec será de grande importância para o LabSEC, uma vez que diversos projetos se beneficiarão do desenvolvimento deste módulo de funções para PHP. Não somente isso, mas os novos projetos do laboratório a serem implementados em PHP já podem ser projetados para utilizar o módulo.

Durante o desenvolvimento do trabalho, foi escrito um artigo sobre o projeto, o qual foi submetido e aceito no Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais 2007 (SBSeg 2007). O simpósio foi realizado na cidade do Rio de Janeiro - RJ, entre os dias 27 e 31 de agosto de 2007. O artigo, intitulado "*Uma implementação de infra-estrutura em Chaves Públicas Temporal*", foi apresentado no Workshop de Trabalhos de Iniciação Científica de Graduação (WTICG), no dia 28. A publicação deste artigo é muito importante para a divulgação do trabalho realizado para o meio acadêmico. O artigo encontra-se nos apêndices deste trabalho.

Com o passar do tempo, novas necessidades surgirão, e modificações ou acréscimos de serviços à ACT serão necessários, uma vez que essa é uma área de estudo recente, e os primeiros passos no sentido de informatização dos serviços sobre os documentos em papel estão sendo realizados.

7 *TRABALHOS FUTUROS*

- Finalizar a implementação da Autoridade Certificadora Temporal de acordo com o projeto proposto, acrescentando ao protótipo desenvolvido os módulos definidos e não implementados (e.g. Módulo Auditor, Módulo Criador), e as funcionalidades faltantes do Módulo Administrador.
- Melhorar a interface gráfica web desenvolvida no protótipo da ACT, disponibilizando um menu personalizado para as operações mais comum da ACT (e.g. testamentos, licitações públicas) de modo a facilitar a utilização dos serviços providos pela ACT.
- Realizar maiores esforços no acerca do módulo de hardware CSM, como o estudo e a implementação ou o aperfeiçoamento desse módulo.

REFERÊNCIAS

ABOUT PostgreSQL. PostgreSQL, 2007. Disponível em: <http://www.postgresql.org/about>. Acesso em: 02 dez. 2007.

AUSTRALIA, A. M.; CAELLI, W.; LITTLE, P. *Electronic signatures - understand the past to develop the future*. [s.n.], 1995. Disponível em: <http://www.law.edu.au/unswlj/ecommerce/mccullagh.html>.

CUSTÓDIO, R. F. et al. An infrastructure for temporal confidentiality in electronic documents. In: . [S.l.: s.n.], 2006.

DIAS, J. S.; CUSTÓDIO, R. F.; DEMÉTRIO, D. B. Sincronizaçõ segura de relógio para documentos eletrônicos. Natal, RN, n. 2, p. 585 – 598, 2003.

DIAS, J. S. et al. Módulo cifrador de documentos eletrônicos. Gramado, RS, n. 1, 2004.

EXTENSION. Zend Developer Zone, 2007. Disponível em: <http://devzone.zend.com/tag/Extension>. Acesso em: 02 dez. 2007.

FORD, W.; BAUM, M. S. *Flows in Networks*. Upper Saddle River, NJ: Prentice Hall, 2000.

MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. *Criptografia em software e hardware*. São Paulo: Novatec Editora Ltda., 2005.

MVC. Wikipedia, 2007. Disponível em: <http://pt.wikipedia.org/wiki/MVC>. Acesso em: 02 dez. 2007.

OPENSSL. Wikipedia, 2007. Disponível em: <http://pt.wikipedia.org/wiki/OpenSSL>. Acesso em: 02 dez. 2007.

PDT Project. Eclipse.org Foundation, 2007. Disponível em: <http://www.eclipse.org/pdt>. Acesso em: 02 dez. 2007.

PEREIRA, F. C.; CUSTÓDIO, R. F.; NOTOYA, A. E. Protocolo criptográfico para envio de propostas em processos de compra. In: *Simpósio de Segurança em Informática*. São José dos Campos: [s.n.], 2003. p. 100 – 110.

PROGRAMMER'S Reference Guide. Zend Framework, 2007. Disponível em: <http://framework.zend.com/manual/en>. Acesso em: 02 dez. 2007.

RIVEST, R. L.; SHAMIR, A.; WAGNER, D. A. Time-lock puzzles and timed-release crypto. Tech. Rep. MIT/LCS/TR-684 Cambridge, MA, USA, 1996. Disponível em: <http://www.rsa.com/pkcs>.

SCHNEIER, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. 2^a. ed. [S.l.: s.n.], 1995.

SILVA, L. S. *Public Key Infrastructure*. São Paulo: Novatec Editora Ltda., 2004.

SIMON, S. *The Code Book*. EUA: Anchor Books, 1999.

STINSON, D. R. *Theory and Practice. CRC Press.* [S.l.: s.n.], 1995.

STREAMS API. Zend Developer Zone, 2007. Disponível em: <http://devzone.zend.com/manual/streams.html>. Acesso em: 02 dez. 2007.

SUBVERSION. Wikipedia, 2007. Disponível em: <http://pt.wikipedia.org/wiki/Subversion>. Acesso em: 02 dez. 2007.

ZEND Framework. Wikipedia, 2007. Disponível em: <http://pt.wikipedia.org/wiki/Zend_Framework>. Acesso em: 02 dez. 2007.

ZEND_DB: Zend Framework Documentation. Zend Framework, 2007. Disponível em: <http://framework.zend.com/manual/en/zend.db.html>. Acesso em: 02 dez. 2007.

APÊNDICE A – ARTIGO APRESENTADO NO SBSEG 2007

Uma implementação de Infra-estrutura em Chaves Públicas Temporal

Geovani Ferreira da Cruz[1], **Guilherme Steinmann**[1]

[1] Laboratório de Segurança em Computação – Universidade Federal de Santa Catarina (UFSC) – Caixa Postal 476 – 88040-900 – Florianópolis, SC

{geovany,guist}@inf.ufsc.br

Abstract. *Some electronic documents used in applications like public biddings and wills require temporal confidentiality as a security requirement, assuring that the document's content will be only released in a specific date and time. Due to modern computer facilities, the electronic documents tend to be more used than papers documents in these kinds of applications. This paper proposes a new Temporal Certification Authority implementation with improvements in the digital certificate's temporality.*

Resumo. *Alguns documentos eletrônicos utilizados em aplicações como licitações públicas e testamentos requerem temporalidade como requisito de segurança, assegurando que o conteúdo do documento será somente liberado em uma data e hora pré-especificadas. Devido as facilidade da computação moderna, os documentos eletrônicos tendem a serem mais utilizados do que documentos em papel. O objetivo deste artigo é propor uma nova implementação de Autoridade Certificadora Temporal, com aprimoramentos e melhorias na temporalidade dos certificados digitais.*

1. Introdução

Com a presença cada vez mais marcante da informática no cotidiano das pessoas os serviços informatizados passam a lhes proporcionar comodidade, eficiência e redução de custos. Dessa

forma uma maior utilização de documentos digitais torna-se inerente às necessidades cotidianas. As operações computacionais sobre documentos digitais vêm sendo implementadas de modo a fornecer ao usuário a garantia de qualidade dos serviços realizados.

Apesar de estar em ascensão a aceitação e utilização de documentos digitais, há a necessidade de melhorias no sistema, como ressaltam Moreno, Pereira e Chiaramonte (2005) da dificuldade em garantir a segurança das informações. Tanembaum (2003) classifica os problemas referentes à segurança das informações em sigilo, autenticação, não-repúdio e integridade. Técnicas de sigilo evitam o acesso às informações por entidades não-autorizadas. A integridade garante que o documento original não seja alterado. A autenticação assegura a identidade da entidade em questão. O não-repúdio evita que a entidade transmissora negue a autenticidade das atividades efetuadas. Técnicas como funções de hash, assinatura digital e certificação digital, da área de segurança em computação, que garantem os requisitos mencionados e quando utilizados em conjunto fornecem um serviço seguro, assim como nos documentos de papéis.

Tendo em vista a viabilidade da implementação de serviços seguros sobre documentos digitais, existem diversos tipos de aplicações que requerem a segurança dos documentos, especificamente àquelas que necessitem enviar documentos para o futuro. Como exemplo, um testamento é um documento que necessita ser mantido em sigilo e somente pode ter seu conteúdo revelado após o falecimento do criador. Outro exemplo são as provas de vestibulares que somente podem ser divulgadas no dia do vestibular. Fora do mundo digital, enviar um documento para o futuro consiste em criar um documento, assiná-lo, lacrá-lo e mantê-lo em um local seguro até uma data pré-especificada. O documento necessita ser mantido em sigilo por este determinado período de tempo e nem mesmo o criador do documento pode ter acesso ao seu conteúdo. Somente após a data pré-especificada o documento pode ter seu conteúdo revelado. Um terceiro exemplo de aplicação deste tipo são as licitações públicas, onde as propostas enviadas pelos fornecedores são mantidas em sigilo até a data de abertura das propostas. No mundo digital, para que tais aplicações sejam viabilizadas, o requisito de temporalidade deve ser acrescido aos requisitos de segurança a fim de garantir o manuseio de documentos ao longo do tempo.

O objetivo deste artigo é propor uma nova implementação de Autoridade Certificadora Temporal (ACT) que garanta aos usuários o fornecimento de serviços eficientes e seguros, que possibilite o envio de documentos para o futuro e, ao mesmo tempo, possibilite a informatização dos processos que necessitem temporalidade. Primeiramente, será realizado um estudo de infra-estrutura em chaves públicas temporal visando contextualizar a atuação de uma ACT e em seguida será apresentada a proposta. O artigo encontra-se estruturado da seguinte forma: A Seção 2 apresenta a definição do problema descrevendo alguns requisitos de temporalidade e

suas aplicações. A Seção 3 apresenta os conceitos relacionados à área de criptografia e define uma infra-estrutura em chaves públicas temporal, fundamentais para o entendimento da solução do problema. A Seção 4 trata da proposta de implementação da ACT. Ao final são descritos possíveis trabalhos futuros.

2. Definição do Problema

Através de um levantamento bibliográfico realizado pôde-se concluir que existem poucas soluções na área de criptografia temporal em se tratando de temporalidade de documentos eletrônicos. Ainda, dessas poucas soluções pesquisadas, não se tem conhecimento de uma que atenda integralmente a todos os seguintes requisitos: temporalidade, sigilo, autenticação, não-repúdio e integridade.

Uma aplicação do requisito temporalidade de documentos se encontra nos processos de licitações públicas. Este processo se divide em modalidades como leilão, pregão, tomada de preços, concorrência, dentre outras. Uma maior abordagem acerca deste processo pode ser consultada em Pereira (2003). Abaixo, as fases de realização de uma licitação pública:

- Preparação: A entidade pública desenvolve o Edital referente à modalidade de compra desejada e o divulga aos interessados através do meio de comunicação apropriado.
- Habilitação: Os interessados entregam a proposta à entidade pública em um envelope lacrado. Essa proposta contém as especificações do objeto de venda, bem como os demais documentos necessários.
- Julgamento: É realizada uma sessão pública, em data e hora especificadas no Edital do processo, onde são abertas as propostas dos fornecedores. A entidade pública seleciona a melhor proposta, através dos critérios de avaliação divulgados no Edital do processo.

Fazendo uma avaliação deste processo, onde os documentos estão em formato não-digital, os requisitos de segurança mencionados são assegurados. O requisito de temporalidade (acesso ao conteúdo somente na data e hora previstas) é garantido através da realização da sessão pública. A integridade do documento é verificada ao não se encontrar rasuras (modificações, colagens, partes do documento riscadas, rasgos ou sobrescritos) no documento de papel. O não-repúdio e a autenticidade são garantidos através da assinatura esferográfica ou do carimbo do fornecedor. O sigilo, por sua vez, é alcançado através da verificação do lacre do envelope contendo os documentos (uma vez que o lacre esteja intacto é possível garantir que ninguém teve acesso ao conteúdo do documento).

Para que, por exemplo, o processo de licitação pública possa ser realizado digitalmente, pelo menos a mesma credibilidade sobre o processo deve ser mantida em se tratando de segurança. Realizar este processo em meio digital traz várias vantagens às entidades envolvidas neste processo. Para a entidade pública, os custos com a infra- estrutura necessária para a realização da sessão pública (local, funcionários, organização) seriam praticamente eliminados e o processo se torna mais eficiente e prático. Para os fornecedores, eliminar-se-iam os custos com papel e impressão e haveria uma maior garantia contra fraudes do sistema público, uma vez que as propostas seriam avaliadas automaticamente pelo sistema informatizado do processo.

Com os avanços obtidos nos estudos das áreas de criptografia e criptografia temporal, várias técnicas surgiram para garantir a segurança dos documentos digitais. De acordo com Custodio, Dias e Pereira (2006), a utilização de funções hash garante a integridade do documento digital. As assinaturas digitais utilizadas em conjunto com funções hash garantem a autenticidade. O não-repúdio dos documentos é alcançado com o uso da criptografia assimétrica, através da utilização de pares de chaves correspondentes (uma pública e uma privada) para cifrar e decifrar os documentos. O sigilo dos documentos é alcançado através dos algoritmos de cifragem e decifragem. De acordo com Rivest, Shamir e Wagner (1996), a temporalidade pode ser alcançada através do uso de uma Terceira entidade confiável.

Em prol de tornar viável a implementação de operações informatizadas sobre documentos digitais que necessitem enviar documentos para o futuro, com a mesma segurança, ou até maior que nos documentos em papel os requisitos de segurança devem ser considerados. Uma vez que não existem soluções completas nesse sentido, o problema é implementar uma infra-estrutura de componentes que utilizem as técnicas de criptografia e criptografia temporal, que interajam entre si, considerando esses requisitos. Uma Autoridade Certificadora Temporal é a entidade chave desta infra-estrutura e, portanto, necessita de uma implementação segura e eficiente para poder ser utilizada pelas aplicações que a necessitam.

3. Infra-estrutura em Chaves Públicas Temporal

Uma Infra-estrutura em Chaves Públicas Temporal (ICPT) consiste numa infra-estrutura de entidades que fornecem serviços de criptografia temporal. As Autoridades Certificadoras, as Autoridades Certificadoras Temporal, as Autoridades de Datação, os Módulos de Hardware Seguro, os Módulos de Relógio Seguro e os Servidores de Carimbo de Tempo são exemplos de entidades que compõe essa infra-estrutura. Essa ICPT é gerenciada por administradores, que por sua vez são fiscalizados pelos auditores. Nos próximos parágrafos serão apresentadas estas entidades e demais conceitos relacionados a uma ICPT.

Uma Autoridade Certificadora (AC) é uma entidade responsável por assinar digitalmente os certificados digitais de usuários ou mesmo de outras Autoridades, garantindo a veracidade destes certificados. Uma AC provê um conjunto de serviços e procedimentos para possibilitar o uso da certificação digital. No contexto da ICPT, uma AC é responsável por assinar digitalmente o certificado digital da Autoridade Certificadora Temporal. Em outras palavras, a confiança da Autoridade Certificadora Temporal está atrelada à confiança que se tem na AC que assinou seu certificado.

As ACs a serem utilizadas nessa ICPT seguem o padrão PKI X.509 [ITU-T X.509]. O padrão propõe uma hierarquia de ACs organizadas em forma de árvore, onde existe uma AC-raiz pertencente ao maior nível da hierarquia. Ela é responsável por certificar outras ACs, que por sua vez, podem assinar os certificados das entidades, ou podem certificar outras autoridades. Os certificados emitidos por uma AC seguem o formato X.509 [RFC4523].

Com o objetivo de garantir a segurança das chaves criptográficas, um hardware especial pode ser utilizado. Dias, Notoya e Pereira (2004) definem um módulo de hardware seguro, ou Hardware Security Module (HSM) em inglês, como um dispositivo de hardware, com poder de processamento e armazenagem, específico para executar serviços tais como a geração e o armazenamento de chaves, realização de operações criptográficas, em geral, sobre aplicações que exigem um elevado grau de segurança. O HSM gera e utiliza chaves criptográficas não permitindo que entidades externas tenham acesso as mesmas. Caso exista alguma tentativa de violar este hardware, o mesmo destrói todo o seu conteúdo.

Uma Autoridade de Datação (AD) consiste numa entidade confiável responsável por fornecer a data e hora corretas para as demais entidades interessadas, que confiam na imparcialidade desta Autoridade [Dias, Custodio e Demétrio 2003]. É composta basicamente, de um relógio que deve estar sincronizado com uma fonte confiável de tempo.

Algumas aplicações de criptografia temporal exigem um elevado grau de segurança e/ou de desempenho. Tais aplicações podem utilizar um hardware específico, desenvolvido para atender estes requisitos, como um Módulo de Relógio Seguro (CSM). Um CSM é um dispositivo de hardware projetado para executar funções de criptografia temporal tais como carimbo de tempo, geração do par de chaves criptográficas e armazenamento de chaves.

Ainda não existe no mercado um CSM com as características apresentadas. Trata-se de um projeto em desenvolvimento pelo Laboratório de Segurança em Computação (LabSEC) da Universidade Federal de Santa Catarina. Esse projeto trata-se de um dos objetivos da dissertação de mestrado de Juliano Romani que consiste na implementação deste dispositivo de modo a realizar uma sincronização segura de relógio. A Figura 1 mostra um exemplo do serviço de for-

necimento de hora provido por uma Autoridade de Datação através de um servidor de carimbo de tempo, bem como uma possível arquitetura de utilização do CSM.

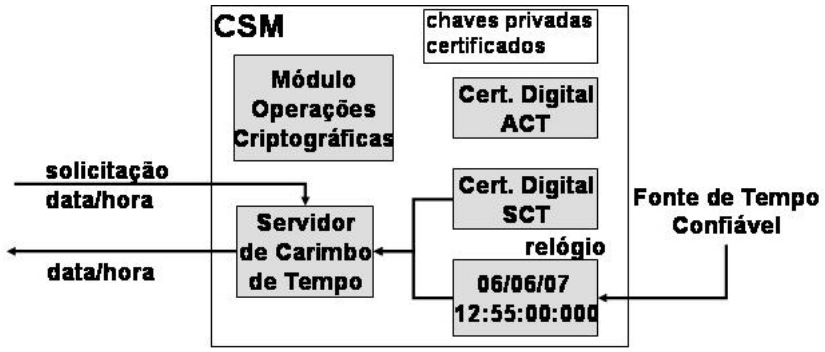

Figura 1. Exemplo de CSM e o serviço de fornecimento de tempo.

A Autoridade Certificadora Temporal é a principal entidade de uma ICPT. Uma ACT é responsável por emitir os certificados digitais temporal, os quais contém uma chave pública que pode ser utilizada pelos usuários da ICPT para cifrarem seus documentos. Esses documentos permanecem cifrados até a liberação (pela ACT) de um outro certificado contendo a chave privada correspondente a chave utilizada para cifrá-los. Uma vez que os usuários da ICPT tenham posse desta chave privada torna-se possível o deciframento do documento.

As ACTs necessitam ter seus relógios fornecendo uma data e hora confiáveis para seus serviços temporais. Esse relógio deve estar sincronizado com um relógio externo confiável, como o de uma Autoridade de Datação.

Portanto, nos serviços fornecidos por uma ACT têm-se a segurança da informação assegurada pelo uso das técnicas de criptografia assimétrica (certificação digital, assinaturas digitais, cifragem e decifragem de documentos) e a temporalidade assegurada pela confiança atribuída a Autoridade de Datação.

4. Trabalhos relacionados:

O trabalho de maior relevância na área de criptografia temporal foi desenvolvido por Rivest, Shamir e Wagner em 1996 [Rivest, Shamir e Wagner 1996]. Eles propõem duas técnicas de implementar a temporalidade de documentos digitais: através de um quebra-cabeça computacional (Time-lock puzzle) e através do uso de uma terceira entidade confiável.

A temporalidade por quebra-cabeça computacional consiste na realização de uma seqüência de operações matemáticas, não escalonáveis e não distribuídas, aonde o tempo total de processamento dessas operações é aproximadamente o tempo que a informação deve permanecer em sigilo. Ao finalizar a seqüência de operações, a informação é descoberta. Torna-se inviável para

o escopo deste projeto uma vez que o método não garante uma precisão no tempo de liberação da informação e porque é necessário um hardware dedicado para processar as operações, ainda desconsiderando o a possibilidade de falhas do hardware.

Uma segunda maneira de se obter a temporalidade é através de uma terceira entidade confiável, a qual se compromete em manter o sigilo da informação até o seu tempo de liberação. A esta terceira entidade dá-se o nome de Autoridade Certificadora Temporal que é a autoridade de qual será realizada a proposta de implementação na próxima seção.

5. Projeto

Para fornecer ao usuário final os serviços temporais, tal como o envio de documentos para o futuro, todos os componentes dessa infra-estrutura em chaves públicas temporal se tornam necessários. Contudo, o projeto de nosso trabalho se limita a uma nova implementação de Autoridade Certificadora Temporal, que é o componente principal da infra-estrutura que irá prover a temporalidade.

Uma infra-estrutura em chaves públicas temporal será utilizada por três diferentes classes de usuários: os administradores, os auditores e os usuários. Os administradores são responsáveis por criar e gerenciar as autoridades certificadoras temporais (ACT). Os auditores são responsáveis por fiscalizar as operações realizadas pelas autoridades e seus respectivos administradores. Os usuários apenas utilizam os serviços providos pela ICPT.

Conhecidos os componentes e os tipos de usuários que utilizam uma ICPT, pode-se fazer um estudo sobre o seu funcionamento. Primeiramente, as diversas autoridades dessa infra-estrutura devem ser criadas e devem ser cadastrados os administradores e auditores responsáveis para elas. Os administradores começam por definir uma política de utilização da autoridade. No caso da ACT, questões como a política de uso, o número máximo de certificados que um usuário poderá requisitar num determinado período de tempo, quais os critérios de aprovação dos usuários e a necessidade ou não de utilizar um CSM. Todas essas operações realizadas pelos administradores podem ser fiscalizadas pelos auditores responsáveis pela autoridade através de relatórios e logs gerados automaticamente pelos sistemas.

Uma ACT irá disponibilizar os certificados temporais através de um diretório público. Os certificados temporais se classificam em dois tipos: os gerados automaticamente pelo sistema e os gerados pelos usuários. Os certificados gerados automaticamente devem atender as necessidades de grande parte dos usuários. Caso o usuário queria algum certificado exclusivo, isto é, algum certificado que contenha uma política específica, este deverá acessar o sistema através de seu cadastro e requisitar o certificado temporal. Caso o usuário não possua cadastro, deverá

cadastrar-se na ACT e aguardar a aprovação de um administrador.

A Figura 2 mostra o processo de enviar documentos para o futuro, o qual consiste em:

1.O usuário acessa o diretório público da ACT e verifica se existe um certificado temporal gerado automaticamente pelo sistema de acordo com suas necessidades;

   1.1.Caso não exista no diretório público um certificado que atenda as necessidades do usuário e o usuário não possua cadastro na ACT, o mesmo efetua seu cadastro;

   1.2.O usuário, uma vez com seu cadastro aprovado, acessa o sistema, requisita a geração de um certificado temporal e submete-o à ACT.

   1.3.A ACT aprova a requisição de certificado temporal gerado pelo usuário e publica o certificado temporal no diretório público;

2.O usuário busca seu certificado digital temporal no diretório público;

3.O usuário cifra seus documentos com o certificado temporal adquirido;

4.O usuário envia os documentos cifrados aos seus destinatários;

5.Na data de liberação do certificado temporal, a ACT publica no diretório público este certificado com a chave privada correspondente à chave pública utilizada pelo usuário pra cifrar os documentos;

6.O usuário e os destinatários podem buscar o certificado com a chave privada dos documentos cifrados no diretório público;

7.O usuário e seus destinatários decifram os documentos e têm acesso ao seu conteúdo.

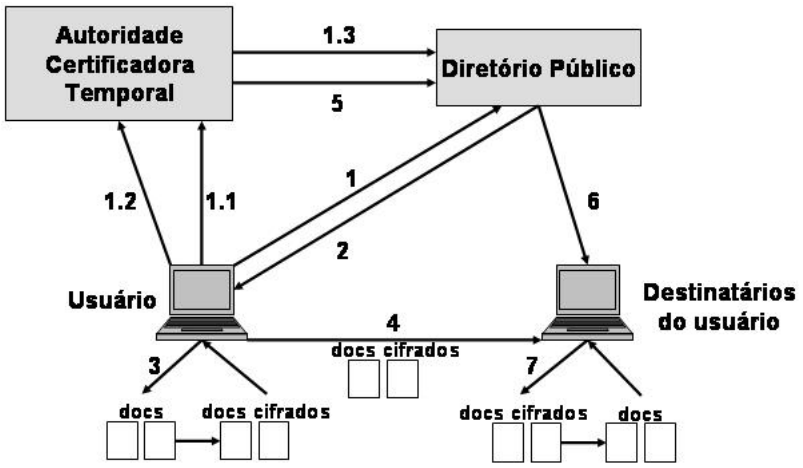

Figura 2. Funcionamento de uma Infra-estrutura em Chaves Públicas Temporal.

A ACT possui um relógio interno, sincronizado com uma autoridade de datação. Ela fica constantemente verificando o relógio e se há algum certificado para liberar. Ao atingir o horário de liberação do certificado temporal a ACT os publica automaticamente.

Um exemplo de uma possível arquitetura de infra-estrutura em chaves públicas temporal pode ser visualizado na Figura 3. Existem três tipos de autoridades certificadoras. A AC-raíz, que possui seu certificado digital auto-assinado, as ACs regionais ou intermediárias (caso da AC-UFSC e da AC-UFPR da Figura 3), que têm permissão de assinar os certificados digitais de outras autoridades e as autoridades finais (AC LabSEC) que podem somente assinar certificados digitais de usuários. Cada autoridade certificadora pode ou não utilizar os serviços de um HSM. No exemplo, a AC-Raiz assinou digitalmente os certificados da AC-UFSC e da AC-UFPR. A AC-UFSC, por sua vez, assinou digitalmente os certificados da AC-LabSEC, da ACT-UFSC, e do servidor de carimbo de tempo da UFSC (SCT-UFSC). A SCT-UFSC fornece os serviços de data e hora à AC-LabSEC e à ACT-UFSC. Várias arquiteturas podem ser formadas, e neste exemplo a SCT-UFSC e a ACT-UFSC compartilham o mesmo dispositivo de hardware, o CSM. A ACT-UFSC é gerenciada por administradores, que por sua vez são fiscalizados por auditores. Os usuários dessa infra- estrutura comunicam-se com a ACT-UFSC através de uma interface WEB que provê os serviços de diretório público, cadastro de usuários e geração de requisições de criação de certificados temporais.

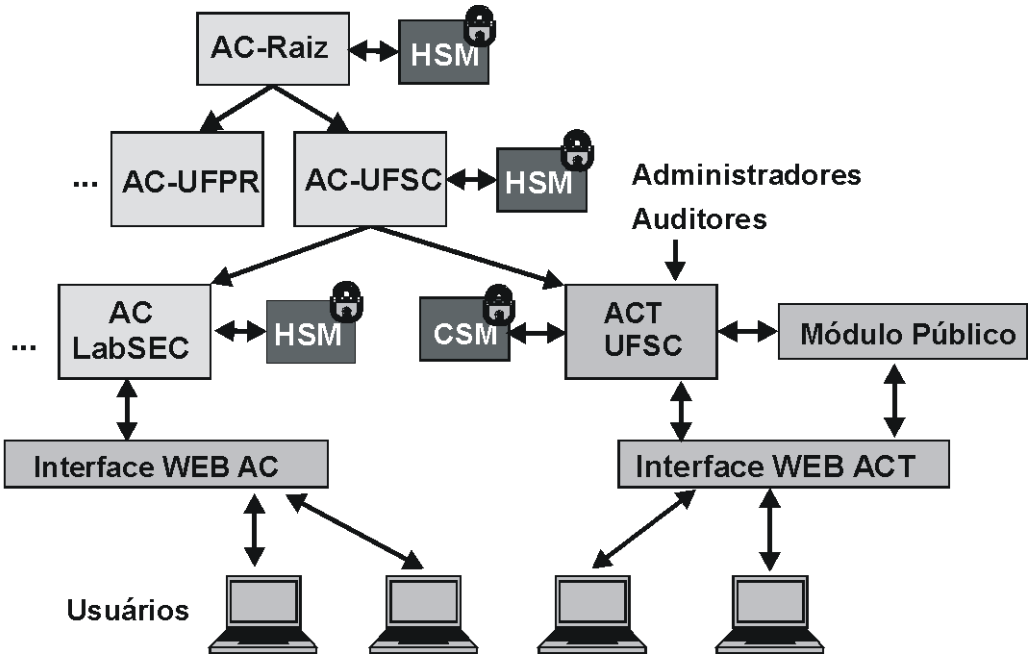

Figura 3. Exemplo de uma possível arquitetura de Infra-estrutura em Chaves Públicas Temporal.

6. Considerações finais e trabalhos futuros

O grande desafio da criptografia temporal é tornar possível a realização, em meio digital, dos serviços realizados sobre os documentos em papel que envolvam temporalidade. A proposta apresentada está em desenvolvimento pelos autores desse artigo, e vem de encontro a esse objetivo. Trata-se do Trabalho de Conclusão de Curso (TCC) dos autores do curso de Ciência da Computação da Universidade Federal de Santa Catarina (UFSC). Várias aplicações como licitações públicas, testamentos, e provas de vestibulares encontram aplicação sobre essa infra-estrutura.

O projeto, já passou pelas fases de levantamento de requisitos e de casos de uso. Atualmente os autores do artigo estão terminando a fase de modelagem e em breve iniciarão os trabalhos de implementação. Para a implementação será utilizada a linguagem PHP e o Zend Framework (um framework MVC para PHP). Em paralelo o mestrando Juliano Romani está desenvolvendo o módulo seguro de relógio que irá fornecer o horário correto para a ACT. Após concluído, o projeto será executado numa máquina do Laboratório de Segurança em Computação (LabSEC) da Universidade Federal de Santa Catarina e disponibilizado para acesso através da internet.

Maiores esforços devem ser realizados acerca do módulo de hardware CSM, como estudo e a implementação ou o aperfeiçoamento desse módulo. Com o passar do tempo, novas necessidades surgirão, e modificações ou acréscimos de serviços dessa infra-estrutura se tornarão necessárias, uma vez que essa é uma área de estudo recente, e os primeiros passos no sentido de informatização dos serviços sobre os documentos em papel estão sendo realizados no momento da publicação desse artigo.

Referências

Custodio, F. R., Dias, J. S., Pereira, F. C., et. al. (2006) “An infrastructure for temporal confidentiality in electronic documents”.

Dias, J. S., Custodio, R. F. e Demétrio, D. B. (2003), “Sincronização Segura de Relógio para Documentos Eletrônicos”, In: Simpósio Brasileiro de Redes de Computadores, 2003, Natal - RN. SBRC 2003, v.2. p.585 - 598.

Moreno, E. D., Pereira, F. D. and Chiaramonte, R. B. (2005), “Criptografia em software e hardware”, Novatec Editora Ltda., São Paulo.

Rivest, R. L., Shamir, A., Wagner, D. A. (1996), “Time-lock puzzles and timed-release crypto”. Tech. Rep. MIT/LCS/TR-684, http://www.rsa.com/pkcs, Cambridge, MA, USA. RSA. 2004.

Tanembaum A. S. (2003), Redes de computadores. Rio de Janeiro: Elsevier.

Dias, J. S., Notoya, A. E., Pereira, F. C., et. al. (2004), "Módulo Cifrador de Documentos Eletrônicos", In: WSeg 2004 - IV Workshop em Segurança de Sistemas Computacionais, In: IV Workshop em Segurança de Sistemas Computacionais, v. 1.

Pereira, F. C. (2003), "Criptografia Temporal: Aplicação Prática em Processsos de Compra" Dissertação (Mestrado em Ciência da Computação) - Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina.