Cryptography Paper Summary

[redacted]

Introduction

The article *Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption* (ASIACRYPT '08) by Hanaoka and Kurosawa describes a CCA-Secure KEM scheme under the CDH assumption in the standard model. Since it's already known how to accomplish this in the random oracle model, the paper is presumably motivated by a combination of distrust in instantiating random oracles and by academic interest in how far we can go without them.

It seems another group found such a scheme independently, and beat Hanaoka and Kurosawa to publication; as a result, H&K emphasize better efficiency (in terms of computation and ciphertext size) as a selling point. Additionally, abstracting away the details of their implementation yields a method of converting a broadcast encryption (BE) system with verifiability into a CCA-secure KEM (and hence a CCA-secure PKE) under the CDH assumption.

Algorithm description

The paper starts with a (previously known) BE system, makes it verifyable (an adversary cannot easily create ciphertexts that cause different users to get different plaintexts), builds it into a one-way KEM, and finally turns the one-way KEM into a CCA-secure KEM under the CDH assumption.

The Naor-Pinkas broadcast encryption scheme

First the BE system. If we want to broadcast messages to a group of people, but want to reserve the ability to exclude up to t of them in the future, then we generate a random polynomial $f(x) = a_0 + a_1 x + \cdots + a_t x^t$ over $\mathbb{Z}_p$ Each person recieves a $(i, f(i))$ pair ($i \neq 0$) as a decryption key. Additionally, t other pairs are made public. To send a message with fewer than t revoked users, we use one or more of these as the "revoked" keys.

In general, anyone who knows $t+1$ distinct points on a polynomial h can reconstruct h completely if $\deg h = t$. Similarly, by exploiting the isomorphism between $\mathbb{Z}_p$ and $G = \langle g \rangle$, a cyclic group of order p, anyone who knows $g^{h(i)}$ for $t+1$ distinct values of i can evaluate $g^{h(x)}$ for any x. (Because the group has order p, it's also a field).

If we choose r at random from $\mathbb{Z}_p$, and publish $pk = (g^r, g^{rf(a_1)}, \ldots, g^{rf(a_t)})$, anyone who knows $(i, f(i))$ for some $i \notin \{a_1, \ldots a_t\}$ can compute $g^{rf(i)}$. This gives them knowledge of $t+1$ distinct points of $g^{rf(\cdot)}$, thereby permitting them to compute aribtrary values of this function – including $g^{rf(0)} = g^{ra_0}$, which we use as a key.

Lacking knowledge of qualified $(i, f(i))$ pair (that is, one for which $i \in \{a_1 \ldots a_t\}$), means computing the value of $g^{rf(0)}$ is provably difficult. However, we're going to make a few changes before we start proving security statements.

Making NP verifiable

The next step is to turn this into a verifiable BE system; that is, make it so the people receiving messages can verify that everyone else who decrypts it gets the same result. In its current form, a user cannot tell if

the ciphertext $(C_0, C_1, \ldots, C_t)$ is "valid", because he'll still get a key from the decryption process even if an adversary has replaced the ciphertext with random bits, in which case the other users would get different keys.

To fix this, we add redundancy to the ciphertext. We up the degree of the polynomial to $2t + 1$ and give everyone *three* $(i, f(i))$ pairs (one of which is randomly selected, with the other two values of i being publicly known). The ciphertext consists of g^r and $2t$ $(x, g^{rf(x)})$ pairs. As before, these pairs use revoked keys supplemented with reserved public decryption keys if fewer than t revoked keys exist. Upon receiving the ciphertext, a person need only use two of his three pairs to find $g^{rf(0)}$, and the result should be the same regardless of which two he chooses.

Building a one-way CCA-secure KEM

The next step is to turn this into a one-way KEM, Π, and here is the first step where the paper gives a formal security proof. We require a TRC hash function, h. Using the verifyable BE with $t = 1$, we construct random polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ over $\mathbb{Z}_p$ of degree three. Our public key is $(G, g, g_0^a, a^{a_1}, g^{a_2}, g^{a_3})$. If someone wishes to send us a message, he takes $r \xleftarrow{\$} Z_p$ and computes g^r, and, using the public key, computes $K = g^{rf(0)}$. The ciphertext he sends us is $(g^r, g^{rf(x_0)}, g^{rf(x_1)})$, where $x_b = h(g^r, b)$. Since we know f, we can verify the ciphertext and compute K.

A proof that this indeed yields a one-way CCA-secure KEM is outlined in the next section, but first we'll finish building the CCA-secure KEM.

The CCA-secure KEM

The final step requires a single hardcore bit hash function $\phi : G \times \{0,1\} \to \{0,1\}$. If we want a k-bit key, then we choose a polynomial with degree $k + 2$. The public key is $pk = (G, g, g^{a_0}, g^{a_1}, \ldots, g^{a_k})$. As before, to generate K we first take $r \xleftarrow{\$} \mathbb{Z}_p$. This time, however, $K = \phi(g^{ra_0})||\cdots||\phi(g^{ra_k})$. Key encryption and decryption are extended in the obvious way. Call this scheme Π'.

Proof summaries

CDH reduces to breaking Π

If there is an adversary A which can obtain K from a valid ciphertext, we can use this adversary to solve the CDH problem. Given (g, g^α, g^β), we want to construct a ciphertext with key $g^{\alpha\beta}$ and ask A to decrypt it for us. To do this, we take $r = \alpha$ (which we cannot directly compute) and give A the ciphertext $(g^\beta, (g^\beta)^{u_0}, (g^\beta)^{u_1})$, where u_b are random values. The two points $(h(g^r, b), u_b)$ fix two points of f, the polynomial we're using as our private key. A third point, (x', u'), is chosen randomly. The final point is $(0, \alpha)$, which prevents us from computing $f(\cdot)$, but is sufficient to compute $g^{f(\cdot)}$.

In order to generate the appropriate public key, we need to compute g^{a_i}, where each a_i is a coefficient of f $(0 \le i \le 3)$. Fortunately, we have

$$f(h(g^r, 0)) = u_0 = \alpha + a_1 h(g^r, 0) + a_2 h(g^r, 0)^2 + a_3 h(g^r, 0)^3,$$

and similarly for $f(h(g^r, 1))$ and $f(x')$. This gives us three equations which we can use solve for g^{a_1}, g^{a_2}, and g^{a_3} by doing linear algebra in the exponent.

When we receive a query (g^a, g^b, g^c) from A, we can return $\perp$ immediately if $g^a = g^\beta$.

If $g^a \neq g^\beta$ but $h(g^a, b) \in \{h(g^\beta, 0), h(g^\beta, 1), x'\}$ we have a problem, since we can't decrypt properly with only three points on $g^{af(\cdot)}$. We just give up and output a random guess. On the other hand, if A does this with non-negligible probability, A just solved the TCR problem for us.

The last possibility is that $g^a \neq g^\beta$, and $h(g^a, b) \neq h(g^\beta, 0), h(g^\beta, 1), x'$. We can't compute $f(\cdot)$, so we can't verify directly. However, because we know can still check for consistency, since if this is a valid ciphertext, we now have five points of $g^{af(\cdot)}$. Using different sets of four of these points, we check to see if the

corresponding polynomials agree at 0. If they do, then let af' denote the polynomial these points determine, and we respond to A's query with $g^{af'(0)}$. It's possible that g^b and g^c do not correspond to points on $g^{af(\cdot)}$. However, the probability that these two (distinct) polynomials agree on x' is extremely small.

If h really is TCR secure, then the two scenarios where we fail to exactly simulate A's expected environment are extremely unlikely. Once A outputs its guess for K, then we output K as well. The formal theorem says that if Π is the one-way KEM and there exists adversaries that run in time τ with advantages ϵ_{cdh} and ϵ_{tcr} in the CDH and TCR(h) experiments, respectively, then

$$\mathrm{Adv}_{\Pi}^{\text{one-way KEM}}(A) \leq \epsilon_{cdh} + 2\epsilon_{tcr} + 3q_D/(p-3),$$

where the A makes q_D queries and runs in time τ.

CDH reduces to breaking Π'

If an adversary can distinguish a key $K = \phi(g^{ra_0})||\cdots||\phi(g^{ra_k})$ from random bits, then a hybrid argument shows that there is an adversary A that can distinguish between $\phi(g^{ra_0})||\cdots||\phi(g^{ra_j})||\$^{k-j}$ and $\phi(g^{ra_0})||\cdots||\phi(g^{ra_{j+1}})||\$^{k-(j+1)}$ for some $j < k$, where $\$^n$ denotes a string of n random bits. Using a method similar to that in the last proof, an HDH-experiment adversary B wishing to decide if some γ it has been given is $\phi(g^{\alpha\beta})$ or a random bit "plants" γ in the appropriate slot of the ciphertext it sends to A.

The security advantage of the hardcore bit function ϕ bounds the HDH problem for ϕ in terms of the advantage of a CDH adversary.

Comments

The authors claim that there systems are practical, and I find it reassuring that we don't have to rely on the random oracle model to have a system based on CDH. However, because the random oracle model seems to give practical results, and because the authors werent quite the first to find a CDH-based CCA-secure KEM in the standard model, I'd have to call the results of this paper incremental. The usefulness would probably depend on performance issues.

That being said, I found it educational and interesting from a mathematical perspective, and suspect that someone better acquainted with BE or other aspects of crypto than myself might be more appreciative of this paper's contributions.

It should also be noted that the paper describes various extensions and variations that I did not mention here. For example, they talked about how to generalize the process of turning verifiable BE schemes into CCA PKEs, and showed how to increase their security bounds under a DDH assumption.